

Article

Not peer-reviewed version

A Novel Secure Prescription System

[Savina Mariettou](#) , [Constantinos Koutsojannis](#) ^{*} , [Vassilis Triantafyllou](#)

Posted Date: 28 February 2025

doi: 10.20944/preprints202502.2301.v1

Keywords: Healthcare Security; Data Protection; Healthcare Cybersecurity; Antibiotic Stewardship; Secure Data Management; Simulated Cyber Attacks



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Novel Secure Prescription System

Savina Mariettou ¹, Constantinos Koutsojannis ^{2,*} and Vassilis Triantafyllou ³

¹ University of Peloponnese, Electrical and Computer Engineering Department, Patras, Greece; s.mariettou@go.uop.gr

² Professor of Medical Physics & Electrophysiology, Director of Health Physics & Computational Intelligence Laboratory, Physiotherapy Department, School of Health Rehabilitation Sciences, University of Patras, Patras, Greece

³ Professor of Network Technologies and Digital Transformation lab, Electrical and Computer Engineering Dpt., University of Peloponnese. Patras, Greece, vtriantaf@uop.gr

* Correspondence: ckoutsog@upatras.gr

Abstract: This research presents a novel system for monitoring antibiotic consumption, addressing the critical need for transparency and accuracy in data management within healthcare settings. The objective is to enhance the monitoring process while ensuring robust security measures. The system's user interface was developed using HyperText Markup Language (HTML) and Cascading Style Sheets (CSS), with Hypertext Preprocessor (PHP) managing database interactions and overall functionality. Security protocols implemented include Transport Layer Security (TLS) 1.3 and 1.2 with Forward Secrecy (FS) to guarantee secure communications. A validation mechanism enforces the use of Hypertext Transfer Protocol Secure (HTTPS) across all URLs, complemented by a 256-bit Elliptic Curve Cryptography (ECC) Secure Sockets Layer (SSL) certificate. The effectiveness of these security measures was evaluated through tests simulating unauthorized access, Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), and SQL injection attacks, demonstrating the system's resilience against various cyber threats. Furthermore, integrating machine learning techniques in Python is proposed to enhance the detection capabilities against SQL injection, thereby fortifying system security. Ultimately, this system aims to optimize hospital resource management, ensuring accurate monitoring of antibiotic consumption and contributing to sustainable healthcare practices.

Keywords: healthcare security; data protection; healthcare cybersecurity; antibiotic stewardship; secure data management; simulated cyber attacks

1. Introduction

The healthcare sector is at the forefront of data digitization, with technologies such as electronic health records, cloud platforms, and e-prescribing being leveraged to store, process, and analyze data [1]. This system enhances efficiency and improves healthcare service quality, contributing to a more reliable and effective system of care [2]. However, the rise of antimicrobial resistance has emerged as one of the ten most critical global public health threats, causing 5 million deaths annually, of which more than half a million occur in Europe and Central Asia [3]. Antimicrobial resistance arises when microorganisms resist the drugs designed to combat them, rendering infections increasingly difficult to treat. Misuse and overuse of antibiotics, particularly in cases of unnecessary prescriptions or non-prescription usage, exacerbate the spread of these "superbugs". If left unchecked, AMR could account for up to 390.000 deaths annually by 2050 [4].

Despite advances in healthcare technology, challenges persist in safely integrating health services. Many hospitals still lack integrated information systems, resulting in the entire process, from diagnosis to treatment, being recorded on paper rather than digitally [5]. This can be confusing,

particularly in vulnerable populations, such as elderly people with chronic conditions, for whom safe and effective sharing of medical data during treatment remains a major concern [6]. In parallel, the rational use of antibiotics is a pressing issue, with studies showing excessive misuse in regions where antibiotics are readily available without prescriptions or where health systems lack strict monitoring protocols [7]. Addressing these dual challenges requires a multifaceted approach that combines robust system security with education and strategies for safe antibiotic use [8].

The proposed simulation-based information technology system offers a unified and secure platform to monitor and optimize antibiotic use, meeting these needs, ensuring transparency in antibiotic consumption, implementing strict data protection measures, and covering the process from diagnosis to treatment. It can serve as a foundation for creating national policies for antibiotics monitoring and infectious disease management worldwide, as many hospitals opt for paper recording and traditional storage for safety reasons.

System security and data integrity have been persistent challenges since 2010 [2]. However, practical solutions, particularly in Greek Health Care Institutions, will not be fully implemented until 2024 [5]. Encryption techniques once deemed reliable are now increasingly vulnerable to cyber threats, highlighting the urgent need for more resilient security frameworks in healthcare [2]. Additionally, proper medication use remains a critical concern, especially in systems where health professionals lack access to unified medical records, making diagnoses and treatments more complex [6].

Antibiotics are essential in treating bacterial infections, as they kill or inhibit their growth. However, improper use, such as unnecessary prescriptions or prolonged durations, has led to the alarming rise of antibiotic resistance. Implementing a simulation-based application system can support healthcare professionals in prescribing antibiotics appropriately, ensuring safer use and reducing resistance [9].

This article presents a secure prescription system that tracks antibiotic consumption and rational use while addressing system vulnerabilities. Its design aims to enhance the diagnosis and monitoring of patient's health. At the same time, it highlights the importance of security in Medical Informatics, through advanced cryptographic techniques and mechanisms, ensuring data protection and compliance with modern health challenges. In Section 2, we present the design and functionality of the proposed system. Section 3 presents the server infrastructure, the system's security measures, and the enhancements we implemented to improve resilience with novel security features. Section 4 evaluates the system's resilience against cybersecurity threats by testing it under four simulated attack scenarios: unauthorized access attempts, Denial-of-Service (DoS) attacks, Distributed Denial-of-Service (DDoS) attacks, and Structured Query Language (SQL) injection attempts. These tests assess the system's ability to withstand real-world security challenges. Section 5 focuses specifically on SQL injection attacks, proposing an enhancement through machine learning to improve detection and system efficiency. Section 6 concludes with a summary of our findings, and Section 7 outlines potential directions for future work.

2. Prescription System Design and Functionality

Infrastructure encompasses the physical structures, facilities, and organizational systems that form the foundation of a nation's economy, health, and security [10]. In healthcare, the adoption of electronic clinical prescription systems has opened opportunities for more effective and timely monitoring of medication management, as seen in aged care initiatives. For instance, the National Aged Care Medication Roundtable demonstrated how leveraging medication administration datasets can enhance safety and decision-making in real-world contexts [11]. Similarly, programs like the Agency for Healthcare Research and Quality (AHRQ) Safety Program for Improving Antibiotic Use underscore the value of structured approaches in utilizing clinical data and decision-making frameworks to optimize antibiotic stewardship while addressing gaps in implementation across diverse healthcare settings [12].

Building on these insights, our system tackles key challenges such as inappropriate antibiotic use and the absence of real-time monitoring. It shifts the focus from theoretical approaches to practical, actionable solutions that enhance prescribing accuracy and patient safety. The system is based on a client-server architecture, where infectiologists and physicians (clients) interact with a web-based interface, as a central server handles processing and communication with the database. The system integrates modern web technologies, including HTML and CSS for user interface design and PHP for database connectivity, to create, scalable, and secure infrastructure. The development process was structured into four distinct stages, as detailed below:

Stage 1: Form Design and Collaboration

In the initial phase, we focused on designing the fill-in form in collaboration with the surveillance team antibiotics. The form facilitates the systematic registration of patient information, capturing their feedback, treatment progress, and responses to prescribed medications. What sets this system apart from a simple data collection form is that it includes an additional verification process by the attending physician and approval by a virulence specialist at a two-week interval.

An important feature of the system is the approval workflow for ongoing treatment. Specifically, if the patient requires continued therapy, approval must be granted within 14 days [13]. Otherwise, the treatment should be discontinued, and the patient's infection progress must be reassessed. Furthermore, the expert physician can recommend or restrict specific antibiotics, based on clinical data and usage guidelines. Notably, research has shown that the duration of antibiotic treatment can be reduced if the patient shows clinical improvement. Specifically, community-acquired pneumonia (CAP) can be treated in 3 to 5 days, acute exacerbation of chronic obstructive pulmonary disease (AECOPD) in 3 to 6 days, hospital-acquired pneumonia (HAP) in 7 days, and streptococcal pharyngotonsillitis in 4 to 5 days. However, these recommendations are still being studied, and further investigations are needed to confirm their effectiveness [14].

Stage 2: Database Integration

The second stage involved connecting the system to a local MySQL database via PHP. The final structure of the database is also shown in Figure 1. This point allowed for secure storage and further processing of patient data for research purposes. During this phase, we also began monitoring the system for potential cyberattacks, laying the groundwork for robust system security.

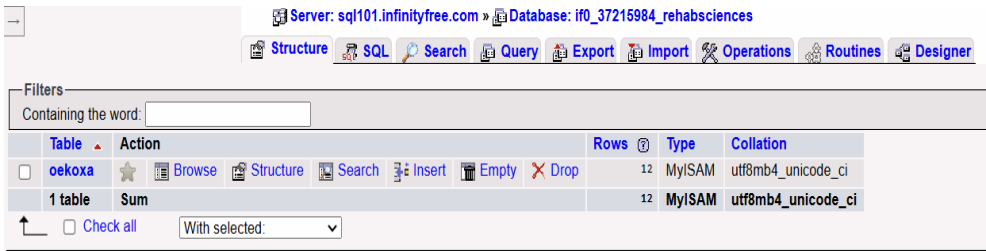


Figure 1. The database structure.

Stage 3: Server Setup and Accessibility Expansion

We built a dedicated server in the third stage to expand system accessibility. This development expanded system accessibility while identifying and mitigating potential security risks and threats to the system infrastructure.

Stage 4: Security Mechanism Implementation

The final and most critical stage involved implementing a specialized security mechanism. This included password protection for the surveillance team tasked with ensuring the proper and responsible use of antibiotics. This security mechanism safeguards the integrity of the prescription system and restricts access to authorized personnel only.

After completing the design and implementation phases, the information technology system was systematically structured into four distinct stages, as illustrated in Figure 2. These stages include

the initial design, local infrastructure development, the transition to remote hosting for broader accessibility, and the robust security system deployment.

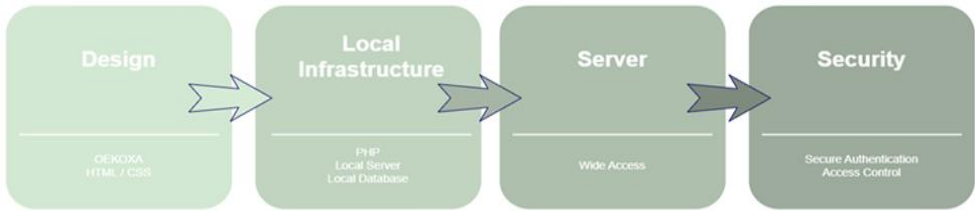


Figure 2. A visual representation of the four stages of the IT system design (Design, Local Infrastructure, Server, and Security).

The fill-in form design incorporated several mandatory fields to ensure accurate and complete data registration. These fields included identifying the hospital, the initial diagnosis date, and, where applicable, approval details from an infectiologist.

12/1/24, 4:00 PM

FORM OERKOA

Surveillance Team for Antibiotic Consumption and Appropriate Use
Hospital Infection Committee

JUSTIFIED PRESCRIPTION OF "PROTECTED" ANTIBIOTICS

Clinic: *

Hospital ID: *

Multidrug-resistant pathogens responsible for nosocomial infections

Gram - Negative

☐ Klebsiella pneumoniae
☐ Pseudomonas aeruginosa
☐ Acinetobacter

Antibiotics under Supervision

☐ Colistin Norma PD.S. Inhaler N
☐ Colistin Norma 1000000/vial
☐ Meropenem 1000MG/vial
☐ Zavisella PD.C.S. INF
☐ INVANZ 16/VIAL LVD INjectable
☐ Fomicyt PD.Sol inf 40mg/ml

DIAGNOSIS - DOSAGE

Diagnosis

Date

Dosage

Duration of Treatment

PRESCRIPTION JUSTIFICATION

Antibiogram (Attachment) No file chosen

A. COMMUNITY-ACQUIRED INFECTION

Attending Physician's Recommendation - Justification:

B. HEALTHCARE-ASSOCIATED INFECTION

Antibiogram (Attachment) No file chosen

☐ Hemodialysis Patient with Severe Sepsis, Septic Shock
☐ Patient with:
• Hematologic Malignancy

12/1/24, 4:00 PM

FORM OERKOA

- Bone Marrow or Solid Organ Transplant
- Recent Hospitalization
- Severe Sepsis / Septic Shock

☐ Neutropenic Patient with Known Colonization by Carbapenemase-Producing Pathogen
Attending Physician's Recommendation - Justification:

☐ Antibiogram

C. HOSPITAL-ACQUIRED INFECTION

☐ Hospital-Acquired Severe Sepsis, Septic Shock (=48 Hours Post-Admission or Earlier if There Has Been an Interventional Procedure in the Hospital)
☐ Hospital-Acquired Infection in Severely Ill Patient (ICU) or Immunocompromised Patient with Hematologic Malignancy / Transplant with Documented Infection or Colonization Within the Last Three Months by Carbapenemase-Producing Pathogen
☐ Hospital-Acquired Infection (Without Sepsis or Septic Shock) When There Has Been Prior Administration of Carbapenems or Colistin During Hospitalization
Attending Physician's Recommendation - Justification:

ATTENDING PHYSICIAN

Signature

APPROVAL BY INFECTIOLOGIST

Signature

Document Date: *

mm/dd/yyyy

Approval Date: *

mm/dd/yyyy

APPROVAL FOR TREATMENT CONTINUATION +14 DAYS

Approval by Infectiologist

Signature

Date: *

mm/dd/yyyy

ATTENDING PHYSICIAN'S INSISTENCE DESPITE NEGATIVE RECOMMENDATION FOR ANTIBIOTIC USE BY INFECTIOLOGIST

Decision by the Head of the Hospital Infection Control Committee

Signature

Date: *

mm/dd/yyyy

Figure 3. (a) Form for Justification of Prescription of "Protected" Antibiotics and Monitoring of Usage (First Part); (b) Form for Justification of Prescription of "Protected" Antibiotics and Monitoring of Usage (Second Part).

3. Server and Security

Following extensive research on healthcare safety systems, the need to strengthen the protection of medical data and the lack of experimental tests assessing the durability of many safety systems has been identified [15]. In response to this research, we have tested the system we created to monitor antibiotics and added additional safety measures to make it more resilient. This process followed the course that we shall analyze hereafter.

This system was tested locally via XAMPP to ensure the website functions properly. Then, hosting on a server was selected via Vista Panel, where the necessary adjustments were started. The system initially provided an RSA 2048-bit (SHA256withRSA) SSL Certificate, which was evaluated for security and performance. After the analysis, the decision was made to switch to an ECC-based SSL certificate for better protection and performance.

At this point, it is important to note that ECC was chosen as an efficient alternative to RSA. ECC offers stronger security with a smaller key size than traditional systems, such as RSA [16]. It provides the same security level with significantly smaller keys, thus reducing the demands on computational resources and improving the speed of cryptographic calculations [17]. Using ECC, a system can operate with limited resources, requiring up to 10% of the bandwidth and storage space that the RSA would need. This efficiency makes ECC ideal for applications where speed and resource savings are critical, such as encrypting embedded medical images [16], as in our system for tracking drugs with Antibioqram, which needs to be attached to our database. ECC is a form of public key cryptography that offers high security with smaller keys, compared to traditional methods such as RSA [17].

Elliptic curves follow the equation:

$$y^2 = x^3 + ax + b$$

(1)

The equation $y^2 = x^3 + ax + b$ where the constants a and b determine the curve’s properties. In cryptographic applications, these parameters are carefully chosen to ensure the curve has no singular points, which is crucial for security.

The process began with generating the ECC Private Key and Certificate Signing Request (CSR) using OpenSSL on Windows. The CSR contained the necessary information to issue the new security certificate. The CSR was then submitted to a Certificate Authority (CA) via an SSL tool to provide the new ECC SSL Certificate. Domain ownership was verified through Canonical Name (CNAME) verification, a process that took some time to complete.

After CA provided the new certificate, we installed it via the Vista Panel, in the SSL/TLS section. The TLS protocol ensures data security and integrity. TLS 1.2 introduced improvements, such as using SHA-256, advanced encryption suites with elliptic curves, and greater flexibility in signature algorithms, enhancing communication protection [18]. We replaced the Private Key and Certificate with the new versions based on Elliptic Curve Cryptography. Once this process was completed, it was necessary to check the installation to confirm a successful transition from RSA to ECC.

The audit was conducted via the Qualys SSL Labs platform, where it was determined that the website was now operating with the new ECC SSL Certificate. The migration to Elliptic Curve Cryptography provided improved security, faster encryption, and higher performance, ensuring that the website uses modern and efficient encryption methods. These upgrades helped transition the system from an RSA to an ECC certificate, improving security and performance.

As far as the security of our database is concerned, we noticed that it was as we can see in Figure 4. Specifically, it uses the secure protocols TLSv1.2 and TLSv1.3, OpenSSL 1.1.1k (FIPS) for encryption, MariaDB 10.6.19 for data management, and InnoDB 10.6.19 for integrity and transactions.

Variable	Value
tls_version	TLSv1.2,TLSv1.3
version_ssl_library	OpenSSL 1.1.1k FIPS 25 Mar 2021
version	10.6.19-MariaDB
innodb_version	10.6.19

Figure 4. Information schema of the MySQL security.

4. Threats and Vulnerabilities: Security Testing through Simulated Attacks

The prescription system was tested against four types of attacks. We theoretically had access to a hospital's closed network in all the attacks. The attacks tested on my system include unauthorized access (Brute Force Attack), Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, and SQL Injection attempts. Below, we will explore each attack in detail, analyzing their methods, potential impact, and the results observed during the tests.

As part of the testing, Docker was used to perform the required tasks within isolated containers, ensuring the accuracy and security of the results. Docker is an open-source technology that allows

the creation and management of isolated, portable environments known as containers. These containers include the software and its dependencies, ensuring consistent performance regardless of the underlying system [20]. Through Docker, the development of replicable environments is simplified, making it ideal for malware analysis, vulnerability testing, and other applications in the cybersecurity field [20].

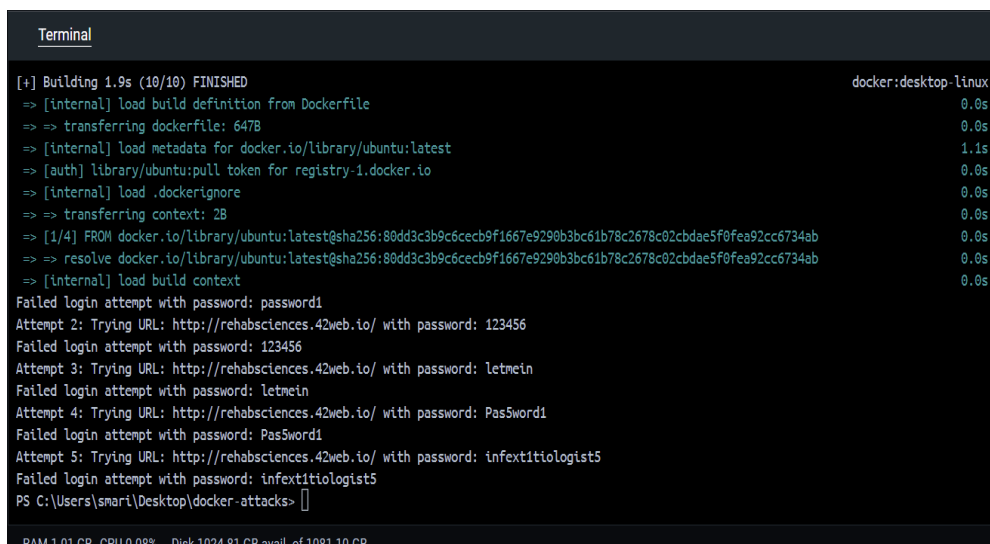
Docker security is based on three main aspects: process isolation, kernel rule enforcement, and network security. It uses Linux kernel features such as namespaces and groups for isolation, while mechanisms such as SELinux, AppArmor, and Seccomp enhance the protection of the host. In the network domain, security is enhanced through TLS for secure image distribution and Docker Content Trust for verifying signed images. However, non-secure options, including a registry, can reduce protection [21]. For this reason, we verified that TLS is enabled in our project, as mentioned in the text. We ensured that Docker Daemon was not publicly accessible and implemented TLS authentication to prevent unauthorized access [22]. Finally, containers are tightly integrated into the host operating system, offering higher performance than virtual machines (VMs) and focusing on the relationship between the host and the container [21].

4.1. Unauthorized Access (Brute force attack)

A Brute Force Attack involves systematically trying all possible combinations of characters until the correct password is found. This attack is perilous when the attacker has additional information or has observed patterns in passwords, such as common passwords or variations of the same password. It exploits weaknesses in easily guessable passwords, and defending against it requires mechanisms like multi-factor authentication and stronger access restrictions [23].

To assess the security of our system, a group of individuals tested the unauthorized access attack, involving unauthorized access, by manually attempting to decode the password. The attempts were then replicated via a script, as shown in Figure 5. The script tried several common passwords, including 'password1', '123456', 'letmein', 'Pas5word1', and 'infe1xt1tologist5', resulting in failed login attempts. The script was executed locally via Docker inside a container created specifically for this purpose.

The attack on our system was unsuccessful, as the system remained fully secure, and no data breach occurred. Had the attack been successful, it could have led to unauthorized access, exposure of sensitive data, and disruption of normal system operations. This highlights the critical need for implementing robust security mechanisms, including multi-factor authentication, strong password policies, and comprehensive access restrictions, to safeguard systems against similar threats in the future.



```

Terminal
[+] Building 1.9s (10/10) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 647B                               0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest   1.1s
=> [auth] library/ubuntu:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                     0.0s
=> [1/4] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 0.0s
=> => resolve docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 0.0s
=> [internal] load build context                                  0.0s
Failed login attempt with password: password1
Attempt 2: Trying URL: http://rehabsciences.42web.io/ with password: 123456
Failed login attempt with password: 123456
Attempt 3: Trying URL: http://rehabsciences.42web.io/ with password: letmein
Failed login attempt with password: letmein
Attempt 4: Trying URL: http://rehabsciences.42web.io/ with password: Pas5word1
Failed login attempt with password: Pas5word1
Attempt 5: Trying URL: http://rehabsciences.42web.io/ with password: infe1xt1tologist5
Failed login attempt with password: infe1xt1tologist5
PS C:\Users\smari\Desktop\docker-attacks>
RAM 1.01 GB CPU 0.08% Disk 1024.81 GB avail. of 1081.10 GB

```

Figure 5. Brute-force attack simulation on a Docker container to detect unauthorized access.

To explore this attack further and its link to the reckless use of antibiotics, it becomes clear that unauthorized access to falsified prescription data presents a significant risk for the health sector [24]. The misuse of antibiotics, whether through incorrect dosages or unauthorized distribution channels, accelerates microbial resistance. In regions with weak regulatory frameworks, the uncontrolled circulation of antibiotics further exacerbates the problem [25].

4.2. Denial of Service (DoS) Attacks

A Denial of Service (DoS) attack aims to exhaust resources such as CPU, memory, and network, causing disruptions to applications and system infrastructure, resulting in performance failure or application crashes [26]. More in detail this attack is a common type of network attack, where the attacker floods the system with excessive traffic, overwhelming its resources. This results in delays or complete unavailability of the healthcare system, preventing legitimate users from accessing critical healthcare information. As part of the Security Attacks in the Network Phase, it targets the network infrastructure, impairing communication and access to essential services [23].

We executed this attack in our system, and as Figure 6 shows, there were 62 attempts in the first test, 451 in the second, and 155 in the third, all classified as DDoS attacks. It is worth noting that as the number of attacks increases, so does the number of users trying to access the system.

The tests were conducted using the Siege tool, executed within a Docker container to ensure consistent testing conditions and isolate the tests from interference with live systems. The Docker environment provided a controlled setup, allowing us to simulate concurrent users and analyze the system's performance under various loads.

This attack concludes that our system remained operational, as indicated by the "availability," which is 100%. The results in Figure 6 highlight the following:

- Response times of 0.15 seconds, 0.14 seconds, and 0.13 seconds;
- Transaction rates of 14.73, 11.70, and 5.21 transactions per second, respectively;
- Concurrency levels of 2.15, 1.62, and 0.70 simultaneous connections;
- Total data transferred: 0.42 MB, 3.13 MB, and 1.26 MB;
- Throughput values of 0.10 MB/s, 0.08 MB/s, and 0.04 MB/s;
- All transactions (62, 451, and 155) were successfully executed without failure, confirming the system's ability to handle increased requests efficiently.

These results validate the system's robustness, showcasing its ability to maintain 100% availability and stable performance, even during simulated DDoS attack scenarios.


```

^C
{
  "transactions": 62,
  "availability": 100.00,
  "elapsed_time": 4.21,
  "data_transferred": 0.42,
  "response_time": 0.15,
  "transaction_rate": 14.73,
  "throughput": 0.10,
  "concurrency": 2.15,
  "successful_transactions": 62,
  "failed_transactions": 0,
  "longest_transaction": 0.54,
  "shortest_transaction": 0.13
}
root@32ccb4ecd16e:/# ^C
root@32ccb4ecd16e:/# ^C
root@32ccb4ecd16e:/# siege -c 5 -t 1M -d 1 -v http://rehabsciences.42web.io/
^C
{
  "transactions": 451,
  "availability": 100.00,
  "elapsed_time": 38.55,
  "data_transferred": 3.13,
  "response_time": 0.14,
  "transaction_rate": 11.70,
  "throughput": 0.08,
  "concurrency": 1.62,
  "successful_transactions": 451,
  "failed_transactions": 0,
  "longest_transaction": 1.14,
  "shortest_transaction": 0.12
}
root@32ccb4ecd16e:/# ^C
root@32ccb4ecd16e:/# ^C
root@32ccb4ecd16e:/# siege -c 5 -t 30s -d 1 -v http://rehabsciences.42web.io/
{
  "transactions": 155,
  "availability": 100.00,
  "elapsed_time": 29.76,
  "data_transferred": 1.07,
  "response_time": 0.13,
  "transaction_rate": 5.21,
  "throughput": 0.04,
  "concurrency": 0.70,
  "successful_transactions": 155,
  "failed_transactions": 0,
  "longest_transaction": 0.15,
  "shortest_transaction": 0.13
}
root@32ccb4ecd16e:/#

```

Figure 6. Simulation of DoS and DDoS attacks on a Docker container to evaluate system resilience.

4.3. Distributed Denial of Service (DDoS) Attacks

DDoS attacks often originate from networks of computers that have been infected with malicious software and are used to attack the target without being immediately detected. The severity of these attacks can be enormous, as they are often difficult to trace back to their source and can last for extended periods, disrupting the functionality of the network or system [23].

The previous section, Denial of Service (DoS) Attacks, provided a comprehensive analysis of the attack, including its behavior and impact. The DDoS attack can be observed in Figure 6 as the second test, where 451 attempts were recorded.

4.4. SQL Injection Attempts

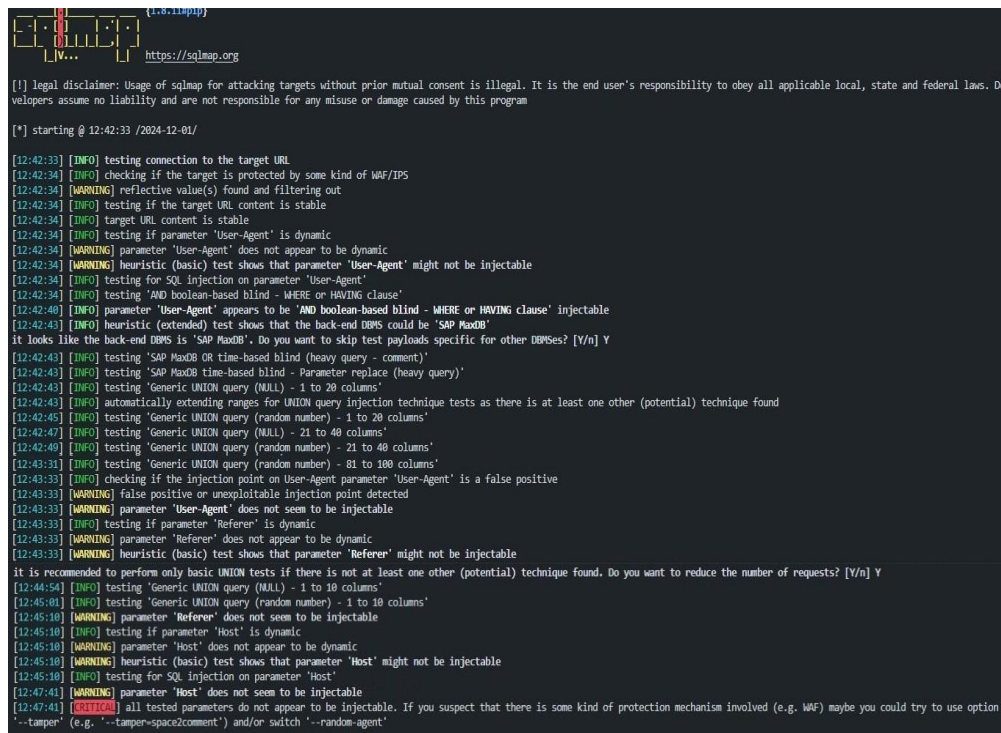
According to Priya et al. (2017), SQL Injection attacks involve injecting malicious SQL statements into a database query to exploit system vulnerabilities. We utilized the SQLMap tool within a Docker container, to simulate and study such attacks. Running SQLMap in a controlled Docker environment ensured consistent results while preventing unintended consequences on other systems. The target web application mimicked common database configurations, allowing SQLMap to execute automated tests on various dynamic parameters. These tests helped identify potential vulnerabilities under realistic conditions.

As part of our research, we conducted an SQL Injection testing scenario. As shown in Figure 7, SQLMap begins the detection process by analyzing dynamic parameters, such as the User-Agent header. Although identified as dynamic, this parameter is ultimately deemed non-exploitable due to possible false positives. The tool then detects SAP MaxDB as the probable backend database management system (DBMS) and recommends focusing on specific payloads for that platform.

Further analysis includes specialized SQLMap payload testing targeting SAP MaxDB, and evaluating additional parameters such as Referer and Host. However, none of these are found to be

vulnerable. Despite testing multiple techniques, including boolean-based blind, time-based blind, and UNION query injection, no exploitable entry points are detected.

SQLMap concludes with a CRITICAL message, indicating that none of the analyzed parameters are vulnerable. This suggests the potential presence of a protection mechanism, such as a Web Application Firewall (WAF), to prevent such attacks. To bypass these defenses, SQLMap suggests using the --tamper=space2comment or --random-agent options.



```
(1.8.11)
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:42:33 /2024-12-01/

[12:42:33] [INFO] testing connection to the target URL
[12:42:34] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:42:34] [WARNING] reflective value(s) found and filtering out
[12:42:34] [INFO] testing if the target URL content is stable
[12:42:34] [INFO] target URL content is stable
[12:42:34] [INFO] testing if parameter 'User-Agent' is dynamic
[12:42:34] [WARNING] parameter 'User-Agent' does not appear to be dynamic
[12:42:34] [WARNING] heuristic (basic) test shows that parameter 'User-Agent' might not be injectable
[12:42:34] [INFO] testing for SQL injection on parameter 'User-Agent'
[12:42:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:42:40] [INFO] parameter 'User-Agent' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[12:42:43] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'SAP MaxDB'
it looks like the back-end DBMS is 'SAP MaxDB'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
[12:42:43] [INFO] testing 'SAP MaxDB OR time-based blind (heavy query - comment)'
[12:42:43] [INFO] testing 'SAP MaxDB time-based blind - Parameter replace (heavy query)'
[12:42:43] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[12:42:43] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[12:42:45] [INFO] testing 'Generic UNION query (random number) - 1 to 20 columns'
[12:42:47] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[12:42:49] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[12:43:31] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[12:43:33] [INFO] checking if the injection point on User-Agent parameter 'User-Agent' is a false positive
[12:43:33] [WARNING] false positive or unexploitable injection point detected
[12:43:33] [WARNING] parameter 'User-Agent' does not seem to be injectable
[12:43:33] [INFO] testing if parameter 'Referer' is dynamic
[12:43:33] [WARNING] parameter 'Referer' does not appear to be dynamic
[12:43:33] [WARNING] heuristic (basic) test shows that parameter 'Referer' might not be injectable
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
[12:44:54] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[12:45:01] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[12:45:10] [WARNING] parameter 'Referer' does not seem to be injectable
[12:45:10] [INFO] testing if parameter 'Host' is dynamic
[12:45:10] [WARNING] parameter 'Host' does not appear to be dynamic
[12:45:10] [WARNING] heuristic (basic) test shows that parameter 'Host' might not be injectable
[12:45:10] [INFO] testing for SQL injection on parameter 'Host'
[12:47:41] [WARNING] parameter 'Host' does not seem to be injectable
[12:47:41] [CRITICAL] all tested parameters do not appear to be injectable. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option
'--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
```

Figure 7. SQLMap detecting potential injection vulnerabilities in the target system.

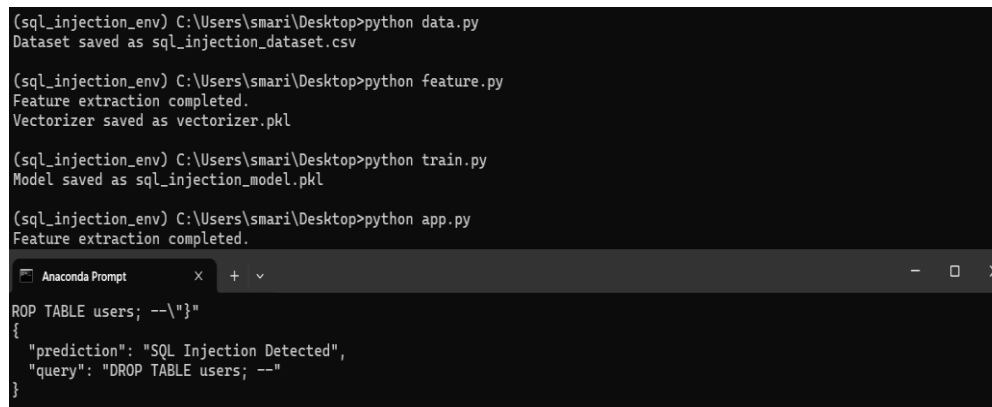
Successful SQL Injection attacks could lead to unauthorized access to sensitive healthcare data, including patient records, compromising privacy and data integrity. This underscores the importance of implementing robust cybersecurity measures to protect medical databases, just as to ensure the proper use of antibiotics. Attackers exploiting such vulnerabilities could alter or delete critical medical data, posing a risk to patient safety and disrupting vital healthcare operations.

Our tests yielded positive results, as we found no exploitable parameters within the target system. The backend database, like SAP MaxDB, has strong security protection. However, these findings emphasize the need for continuous improvements in security measures, as sophisticated attackers may develop advanced techniques to bypass existing defenses.

5. Applying Machine Learning for Securing SQL Injection

While our tests yielded positive results, highlighting the strong security protections of the backend database, they also underscore the importance of proactive measures; thus, to enhance our security framework, we developed an application that integrates machine learning to identify whether SQL injection attempts are safe or malicious. To integrate machine learning into our system, we developed an application to identify whether SQL injection attempts are safe or malicious. Using a simulated dataset we created, we transformed SQL data into numerical representations and trained a Random Forest model for precise classification.

As shown in Figure 8, a vectorizer was employed to convert SQL queries into numerical features, which were saved as 'vectorizer.pkl'. The trained model itself was stored as 'sql_injection_model.pkl'. This setup enables real-time threat detection with high accuracy, allowing the system to distinguish between legitimate queries and potential attacks.



```
(sql_injection_env) C:\Users\smari\Desktop>python data.py
Dataset saved as sql_injection_dataset.csv

(sql_injection_env) C:\Users\smari\Desktop>python feature.py
Feature extraction completed.
Vectorizer saved as vectorizer.pkl

(sql_injection_env) C:\Users\smari\Desktop>python train.py
Model saved as sql_injection_model.pkl

(sql_injection_env) C:\Users\smari\Desktop>python app.py
Feature extraction completed.
```

```
{
  "prediction": "SQL Injection Detected",
  "query": "DROP TABLE users; --"
}
```

Figure 8. SQL injection model training and detection using machine learning.

While testing with real-world examples, such as the query 'DROP TABLE users; --', the system successfully identified the threat and flagged it as an SQL injection attack. This demonstrates the effectiveness of the machine learning approach in enhancing security and mitigating vulnerabilities.

6. Results

Our system provides a safe and effective solution for monitoring antibiotic prescriptions and ensuring their proper use in prescribing and administration. It allows physicians to register and track prescriptions in a structured and secure manner. The form is hosted on a secure HTTPS server, utilizing TLS 1.2 and 1.3 with modern encryption suites to ensure safe data transmission. Following the issuance and integration of the certificate, the system employs Elliptic Curve Cryptography (ECC) with a 256-bit key (Sha384withEcdsa), offering high security at a lower computational cost than traditional RSA encryption.

A comprehensive SSL security assessment confirmed adherence to best practices, supporting Forward Secrecy, mitigating known vulnerabilities, and implementing strong encryption mechanisms. As shown in Figure 9, all cryptographic protocols follow the latest security recommendations, ensuring optimal protection against emerging threats. The cipher suites, TLS AES 256 GCM SHA384 and TLS CHACHA20 POLY1305 SHA256 optimize security and performance. These state-of-the-art algorithms provide high-speed processing and resistance to cryptographic attacks, ensuring that sensitive medical data remains protected. Additionally, the system does not support outdated protocols such as SSL 2.0, SSL 3.0, TLS 1.0, or TLS 1.1, ensuring compliance with modern security standards.

Strong encryption policies enhance the system's resilience and align with industry-leading security frameworks. Compliance checks verified that all connections utilize HTTPS, enhancing data integrity. Furthermore, penetration testing confirmed the system's resilience against external threats, with no detected data breaches or unauthorized modifications.

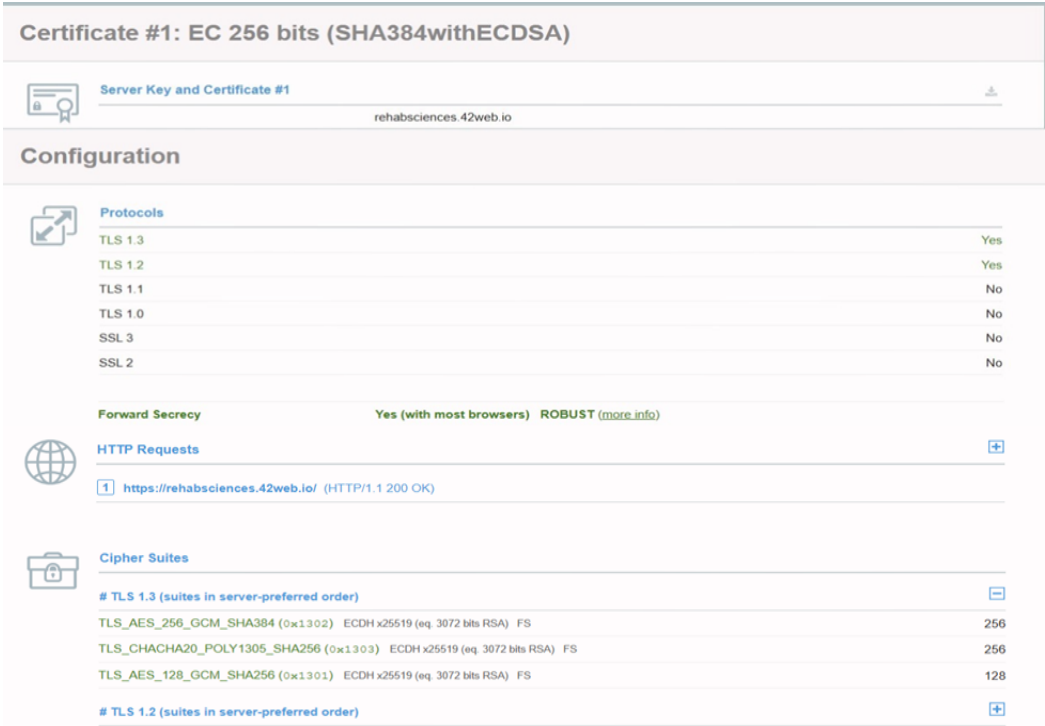


Figure 9. SSL security assessment confirming strong encryption and modern protocol support.

Access is restricted to a single certified physician, as the system is in a trial/research phase. This controlled environment allows us to refine security measures and optimize performance before potential expansion. All files are securely stored within the hospital’s cloud infrastructure, ensuring accessibility and data protection. By facilitating the accurate tracking of antibiotic prescriptions, our system promotes responsible antibiotic use [27], helping to combat antimicrobial resistance while establishing a robust cybersecurity framework for medical data protection. Rather than simply adopting cutting-edge technologies for their own sake, our approach focuses on adapting proven solutions to the real needs and constraints of the healthcare environment, ensuring a high level of security without unnecessary complexity or cost.

7. Conclusion and Future Work

This paper introduced an enhanced prescription system to monitor antibiotic consumption, ensuring transparency, proper data management, and security. Implemented using HTML, CSS, and PHP, the system demonstrated robust resistance to various cyber threats, including unauthorized access, Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, and SQL injection attempts. The results show that our system is reliable for monitoring antibiotics and protecting against cyberattacks. Protecting medical data is essential for safeguarding patient privacy, supporting responsible medication use, and contributing to faster, more effective patient recovery. Currently, the system supports only a single user; however, future enhancements will aim to enable multiple users. This will allow the system to scale and handle a broader range of tasks, such as monitoring the number of consultations and providing data analytics for medical professionals. Future improvements will include implementing multilingual support to accommodate diverse healthcare environments. The system could further evolve by transitioning its database into a blockchain, allowing doctors to form decentralized networks and facilitate the secure addition of new members.

As an extension of our work, integrating blockchain technology could be particularly beneficial for healthcare organizations. Furthermore, the future commercial availability of quantum computing could offer advanced encryption and processing capabilities, further enhancing the system’s security

and efficiency while unlocking new possibilities. Additional security tests, including database-specific attacks and integration with Elliptic Curve Cryptography (ECC) for enhanced network protection, will be crucial for maintaining and improving system resilience. By addressing these future capabilities, the system can become even more secure, scalable, and adaptable, laying the foundation for its broader use and long-term sustainability in healthcare settings.

Author Contributions: Conceptualization, C.K. and V.T.; methodology, S.M. and V.T.; formal analysis, S.M.; writing—original draft preparation, S.M.; writing—review and editing, S.M. and C.K.; supervision, V.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No real-world data were created or analyzed in this study. A synthetic dataset was generated for illustrative purposes, but it does not contain actual research data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
AHRQ	Agency for Healthcare Research and Quality
AMR	Antimicrobial Resistance
CNAME	Canonical Name
CAP	Community-Acquired Pneumonia
CSR	Certificate Signing Request
CSS	Cascading Style Sheets
DDoS	Distributed Denial of Service
DoS	Denial of Service
ECC	Elliptic Curve Cryptography
GCM	Galois/Counter Mode
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
PHP	Hypertext Preprocessor (originally Personal Home Page)
RSA	Rivest Shamir Adleman
SAMaxDB	SAP Maximum Database
SHA	Secure Hash Algorithm Abbreviations
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security

References

1. Mariettou, S.; Koutsojannis, C.; Triantafillou, V. Predicting Coronary Heart Disease through Machine Learning Algorithms. *In Lecture notes in networks and systems*; **2024**; pp. 652–659.
2. SaberiKamarposhti, M.; Ng, K.-W.; Chua, F.-F.; Abdullah, J.; Yadollahi, M.; Moradi, M.; Ahmadpour, S. Post-Quantum Healthcare: A Roadmap for Cybersecurity Resilience in Medical Data. *Heliyon* **2024**, *10*, e31406, doi:10.1016/j.heliyon.2024.e31406.
3. World Health Organization. Control antibiotic misuse or the drugs won’t work, warn WHO experts. Available online: <https://www.who.int/europe/news-room/23-11-2023-control-antibiotic-misuse-or-the-drugs-won-t-work--warn-who-experts> (accessed on 5/11/2024).
4. European Commission. UNGA Political Declaration: A global commitment to fight antimicrobial resistance (AMR). Available online: <https://ec.europa.eu/newsroom/hera/items/852435/en> (accessed on 5/11/2024).

5. Mariettou, S.; Koutsojannis, C.; Triantafillou, V. Security Systems in Greek Health Care Institutions: A Scoping Review Towards an Effective Benchmarking Approach. *Proceedings* **2024**, *92*, 53–60.
6. Wang, Y.-W.; Wu, J.-L. A Privacy-Preserving Symptoms Retrieval System with the Aid of Homomorphic Encryption and Private Set Intersection Schemes. *Algorithms* **2023**, *16*, 244, doi:10.3390/a16050244.
7. English, B.K.; Gaur, A.H. The Use and Abuse of Antibiotics and the Development of Antibiotic Resistance. *Advances in Experimental Medicine and Biology* **2009**, *73–82*, doi:10.1007/978-1-4419-0981-7_6.
8. Javaid, M.; Haleem, A.; Singh, R.P.; Suman, R. Towards Insighting Cybersecurity for Healthcare Domains: A Comprehensive Review of Recent Practices and Trends. *Cyber Security and Applications* **2023**, *1*, 100016, doi:10.1016/j.csa.2023.100016.
9. Salam, Md.A.; Al-Amin, Md.Y.; Salam, M.T.; Pawar, J.S.; Akhter, N.; Rabaan, A.A.; Alqumber, M. a. A. Antimicrobial Resistance: A Growing Serious Threat for Global Public Health. *Healthcare* **2023**, *11*, 1946, doi:10.3390/healthcare11131946.
10. Roumani, Y.; Alraee, M. Examining the Factors That Impact the Severity of Cyberattacks on Critical Infrastructures. *Computers & Security* **2024**, *148*, 104074, doi:10.1016/j.cose.2024.104074.
11. Westbrook, J.I.; Seaman, K.; Wabe, N.; Raban, M.Z.; Urwin, R.; Badgery-Parker, T.; Mecardo, C.; Mumford, V.; Nguyen, A.D.; Root, J.; et al. Designing an Informatics Infrastructure for a National Aged Care Medication Roundtable. *Studies in Health Technology and Informatics* **2024**, doi:10.3233/shiti230996.
12. Tamma, P. D.; Miller, M. A.; Dullabh, P.; Ahn, R.; Speck, K.; Gao, Y.; Scherpf, E.; Cosgrove, S. E. Association of a Safety Program for Improving Antibiotic Use with Antibiotic Use and Hospital-Onset Clostridioides difficile Infection Rates Among US Hospitals. *JAMA Network Open* **2021**, *4*(2), e210235. doi:10.1001/jamanetworkopen.2021.0235.
13. Daneman, N.; Rishu, A.H.; Pinto, R.; Aslanian, P.; Bagshaw, S.M.; Carignan, A.; Charbonney, E.; Coburn, B.; Cook, D.J.; Detsky, M.E.; et al. 7 versus 14 Days of Antibiotic Treatment for Critically Ill Patients with Bloodstream Infection: A Pilot Randomized Clinical Trial. *Trials* **2018**, *19*, doi:10.1186/s13063-018-2474-1.
14. Kuijpers, S.M.E.; Buis, D.T.P.; Ziesemer, K.A.; Van Hest, R.M.; Schade, R.P.; Sigaloff, K.C.E.; Prins, J.M. The Evidence Base for the Optimal Antibiotic Treatment Duration of Upper and Lower Respiratory Tract Infections: An Umbrella Review. *The Lancet Infectious Diseases* **2024**, doi:10.1016/s1473-3099(24)00456-0.
15. Mariettou, S.; Koutsojannis, C.; Triantafillou, V. Artificial Intelligence and Algorithmic Approaches of Health Security Systems: A Review. *Algorithms* **2025**, *18*, 59, doi:10.3390/a18020059.
16. George, N.; Manuel, M. A Secure Data Hiding System in Biomedical Images Using Grain 128a Algorithm, Logistic Mapping and Elliptical Curve Cryptography. *Multimedia Tools and Applications* **2024**, doi:10.1007/s11042-024-19147-2.
17. Farhat, S.; Kumar, M.; Srivastava, A.; Bisht, S.; Rishiwal, V.; Yadav, M. Enhancing E-Healthcare Data Privacy through Efficient Dual Signature on Twisted Edwards Curves Encryption Decryption. *Security and Privacy* **2024**, doi:10.1002/spy2.464.
18. Ivanov, O.; Ruzhentsev, V.; Oliynykov, R. Comparison of Modern Network Attacks on TLS Protocol. International Scientific-Practical Conference Problems of Infocommunications. *Science and Technology (PIC S&T)* **2018**; pp. 565–570. doi:10.1109/INFOCOMMST.2018.8632026.
19. Mouat, A. Using Docker: Developing and Deploying Software with Containers; O'Reilly Media, **2016**.
20. Ahmed, Zeinab, Docker Technology for Small Scenario-Based Exercises in cybersecurity, *Theses and Dissertations* **2023**. 503. Available on https://csuepress.columbusstate.edu/theses_dissertations/503
21. Combe, T.; Martin, A.; Di Pietro, R. To Docker or Not to Docker: A Security Perspective. *IEEE Cloud Computing* **2016**, *3*, 54–62, doi:10.1109/MCC.2016.100.
22. Borglund, N. Security and Application Deployment Using Docker. Degree Project, *Uppsala University*, **2024**. Available online: <https://www.diva-portal.org/smash/get/diva2:1913443/FULLTEXT01> (accessed on 22/02/2025)
23. Priya, R.; Sivasankaran, S.; Ravisasthiri, P.; Sivachandiran, S. A Survey on Security Attacks in Electronic Healthcare Systems. International Conference on Communication and Signal Processing (ICCSP); *IEEE*, **2017**; pp. 691–694. doi:10.1109/ICCSP.2017.8286448.

24. Cesaro, A.; Hoffman, S.C.; Das, P.; De La Fuente-Nunez, C. Challenges and Applications of Artificial Intelligence in Infectious Diseases and Antimicrobial Resistance. *Npj Antimicrobials and Resistance* **2025**, *3*, doi:10.1038/s44259-024-00068-x.
25. Islam, M.F.; Arka, P.B.; Rohman, M.; Hossain, M.S.; Babu, M.R.; Azhari, H.A.; Uddin, M.J. Pooling the Complex Survey Data across the 64 Lower and Middle-Income Countries: A Study on Antibiotic Usage in under-Five Children. *Heliyon* **2024**, *11*, e41470, doi:10.1016/j.heliyon.2024.e41470.
26. Haq, M. S.; Nguyen, T. D.; Tosun, A. Ş.; Vollmer, F.; Korkmaz, T.; Sadeghi, A. -R. SoK: A Comprehensive Analysis and Evaluation of Docker Container Attack and Defense Mechanisms. *IEEE*, **2024**; pp. 4573–4590. doi:10.1109/SP54263.2024.00268.
27. Compton, W.M.; Volkow, N.D. Abuse of Prescription Drugs and the Risk of Addiction. *Drug and Alcohol Dependence* **2006**, *83*, S4–S7, doi:10.1016/j.drugalcdep.2005.10.020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.