OKU Fen Bilimleri Enstitüsü Dergisi
9(2): 994-1011, 2026

OKU Journal of The Institute of Science and Technology, 9(2): 994-1011, 2026

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Bayrak Ele Geçirme Yoluyla Yapılandırılmış Saldırı Güvenliği Uygulaması: Ardışık Sıralı Saldırı Yürütme ve Üç Bayrağın Ele Geçirilmesi

Esra ÇALIK BAYAZIT[1*], Merve ARAÇ[2], Behlül GÜCÜKOĞLU[3], Veysel ER[4], Erva ÖFKELİ[5]

[1] Department of Computer Engineering, Fatih Sultan Mehmet Vakif University, Istanbul, Türkiye
[2,3,4,5] Information Security Technology Program, Fatih Sultan Mehmet Vakif University, Istanbul, Türkiye

[1]https://orcid.org/0000-0002-6813-1037
[2]https://orcid.org/0000-0001-8892-3974
[3]https://orcid.org/0009-0000-1372-6116
[4]https://orcid.org/0009-0002-9753-5847
[5]https://orcid.org/0009-0007-2250-4276
*Sorumlu yazar: ecalik@fsm.edu.tr

Araştırma Makalesi

Makale Tarihçesi:
Geliş Tarihi: 11.12.2025
Kabul Tarihi:27.01.2026
Online Yayınlanma: 16.03.2026

Anahtar Kelimeler:
Bayrak Yakalama
Siber Güvenlik
Bilgi Güvenliği
Saldırı
Zafiyet

ÖZ

Penetrasyon test türlerinden biri olan Bayrak Yakalama (Capture the Flag- CTF), son dönemlerde siber güvenlik alanında her geçen gün daha popüler hale gelen bir yöntem olarak değerlendirme süreçlerinde sıklıkla uygulanmaya başlanmıştır. Kurulan bir senaryo gereği bir tehdit aktörünün saldırısının bir bilgisayar sistemi veya ağının güvenliğini temsilen bayrağı ele geçirme yoluyla simüle edilen bir değerlendirme sürecidir. Bu çalışmada, bir banka sistemi simülasyonun basamakları sunularak siber güvenlik alanında temel öneme sahip olan privilege escalation, SQL injection, command injection, directory traversal, brute force ve steganography saldırı tekniklerinin uygulamalı olarak deneyimlenmesini sağlayacak bir çerçeve sunulmuştur. Ayrıca privilege escalation teknikleri uygulayarak yetki yükseltmeleri, gizli dosyalara ulaşmaları ve Hydra/John the Ripper gibi araçlarla parola kırma süreçlerinin deneyimlenmesi hedeflenmiştir. Bu doğrultuda çalışma, siber güvenlik sektöründe teorik bilginin pratik uygulamaya dönüştürülmesine yönelik yol gösterici bir model sunmayı amaçlamakta; senaryo kapsamında ele geçirilmesi hedeflenen üç bayrağa erişim süreci ise zafiyet istismarının operasyonel gerekliliklerini sistematik bir biçimde ortaya koymaktadır. Ayrıca, katılımcıların ilerleme düzeyleri CTF uygulaması süreci boyunca kullanılan takip scripti aracılığıyla takip edilmektedir. Bu yönüyle çalışma, siber güvenlik alanında eğitim ve farkındalık oluşturmak için etkili, uygulanabilir ve izlenebilir bir simülasyon modeli ortaya koymaktadır.

Structured Attack Security Application via Capture the Flag: Sequential Attack Execution and Capture of the Three Flags

Research Article

Article History:
Received: 11.12.2025
Accepted: 27.01.2026
Published Online: 16.03.2026

Keywords:
Capture the Flag (CTF)
Cyber Security
Information Security
Attack
Vulnerability

ABSTRACT

Capture the Flag (CTF), one of the types of penetration testing, is started to be frequently applied in evaluation processes as a method that has recently become increasingly popular in the field of cybersecurity. Within a predefined scenario, it is an evaluation process that simulates a threat actor's attack by capturing a flag representing a computer system or network security. In this study, a framework is presented that provides practical experience of attack techniques having fundamental importance in the field of cybersecurity including privilege escalation, SQL injection, command injection, directory traversal, brute force, and steganography by presenting the steps of a banking system simulation. Additionally, it is aimed to practice privilege escalation, accessing the hidden files and password-cracking processes using tools such as Hydra/John the Ripper by applying privilege escalation techniques. In this regard, the study aims to offer a guiding

model for transforming theoretical knowledge into practical applications within the cybersecurity sector. The process of capturing the three flags targeted within the scenario systematically reveals the operational requirements of vulnerability exploitation. Furthermore, participants' progress levels are monitored via a tracking script used throughout the CTF application process. In this aspect, the study provides an effective, applicable, and traceable simulation model for training and raising awareness in the field of cybersecurity.

To Cite: Çalık Bayazıt E., Araç M., Gücükoğlu B., Er V., Öfkeli E. Structured Attack Security Application via Capture the Flag: Sequential Attack Execution and Capture of the Three Flags. Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2026; 9(2): 994-1011.

1. Introduction

With the rapid advancement of technology, the demand for competent cybersecurity professionals is increasing across all sectors. In response to this growing need, problem-based learning techniques play a significant role in developing practical skills and in supporting the cultivation of a qualified workforce demanded by the industry (Gardiner et al., 2024). Within this educational framework, Capture the Flag (CTF) competitions have emerged as a critical component of cybersecurity education, as they simulate realistic attack and defense scenarios across various categories. By incorporating challenges with varying levels of difficulty, these simulations encourage participants to develop flexible thinking and adaptive problem-solving skills (Gleeson, 2024).

Since 2015, the Information Systems Audit and Control Association (ISACA) has conducted a research on cybersecurity threats and competency gaps among its members across more than 190 countries. According to ISACA's State of Cybersecurity 2024 report, published on October 1, 2024, data on the time required to fill cybersecurity positions within organizations indicate a persistent shortage of qualified personnel in the industry. As illustrated in Figure 1, 37% of entry level positions and 38% of non-entry-level positions are filled within a 3–6 month period. These percentages represent the highest for both position types, indicating that the process of identifying suitable candidates typically extends to nearly half a year. Furthermore, 27% of non-entry-level roles take longer than six months to fill, highlighting the pronounced and unmet demand for experienced and expert-level professionals. These findings underscore the critical need for skilled specialists within the cybersecurity ecosystem and reveal that extended recruitment processes pose potential risks to organizational security posture (ISACA, 2024).

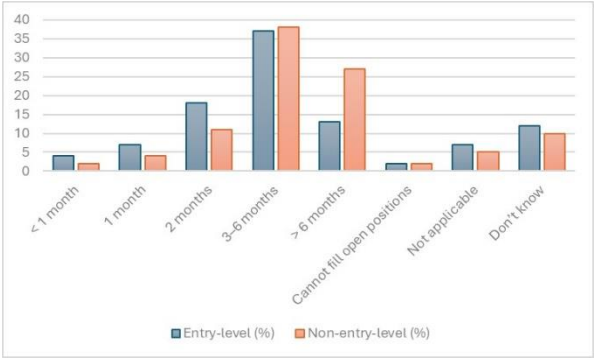

Figure 1. Time to Fill Cybersecurity Positions (ISACA, 2024).

On the other hand, with the advancement of technology, artificial intelligence (AI) applications are increasingly present across various domains. However, the report confirms that AI usage in security operations is still in its early stages and highlights that the involvement of security professionals in the development, deployment, and implementation of AI remains surprisingly low. In participating organizations, security experts are almost entirely excluded from the process of developing policies to govern the use of AI technologies.

Numerous studies and reports have highlighted the persistent shortage of cybersecurity skills in the industry (International Telecommunication Union -ITU, 2024). Even with the availability of advanced technologies and solutions, human decision-making processes remain the most critical factor in preventing security breaches. The report identifies key contributing variables to such breaches, including human error, intentional actions, lack of training, and insufficient awareness.

Addressing the shortage of qualified and competent workforce in the cybersecurity sector, which is a global challenge, requires that students graduating in this field shape their career orientations appropriately. Enhancing students' engagement is crucial by providing learning content that enables them to effectively apply their skills, thereby contributing to the performance improvement demanded by the industry. To properly channel this interest and support their development, problem-based learning methods should be employed, ensuring that students become solution oriented contributors within the sector.

In the CTF scenario presented in this study, a competitive educational activity is introduced in which participants complete various cybersecurity tasks to locate hidden "flags" embedded within computer systems. These flags are concealed behind intentionally created vulnerabilities, requiring participants to exploit these weaknesses in order to reach them. This competitive environment is designed to generate excitement and motivation toward cybersecurity. To achieve these objectives, participants draw upon their knowledge and skills in areas such as programming, network security, and cryptography to solve complex problems. Within this scope, the study involves the development of a simulated banking system, in which specific vulnerabilities are deliberately integrated into the environment, enabling participants to discover these weaknesses and perform penetration attempts by exploiting them. In the CTF simulation that includes security flaws and vulnerabilities, participants are expected to apply techniques such as privilege escalation, SQL injection, command injection, directory traversal, brute force, and steganography, ultimately capturing the three flags. At the end of the challenge, participants are expected to capture the flags hidden within the system and complete a money transfer to their own account.

A review of the literature indicates that the rapidly evolving information systems and digital environment in recent years have significantly increased the demand for professionally trained individuals in cybersecurity.

The study investigates the role of CTF challenges in cybersecurity education, particularly for beginner level penetration testing instruction. By integrating web exploitation, cryptography, steganography, and

digital forensics challenges, the study demonstrates that hands on, vulnerability focused activities such as SSTI, command injection, and kernel exploits significantly improve learners' understanding of core cybersecurity concepts and practical skills. These findings are consistent with prior research highlighting the effectiveness of challenge-based learning approaches in cybersecurity education (Nagare et al., 2025).

In this context, CTF competitions have emerged as an effective learning tool, providing individuals with the opportunity to practically develop their cybersecurity skills, particularly in areas such as cryptography, information security, network security, and reverse engineering. Consequently, CTF competitions stand out as one of the innovative educational tools used to cultivate practical skills in cybersecurity training (Uddin et al., 2021). In a CTF competition, participants attempt to solve technical challenges related to information security by identifying hidden information referred to as flags through the detection of specific vulnerabilities. These flags, often presented in textual form, are used either to earn points or to advance to subsequent stages of the competition. Through this process, participants are able to assess and test their own level of cybersecurity proficiency.

CTF competitions are controlled contests centered on identifying and exploiting security vulnerabilities, in which participants typically undertake both defensive and offensive tasks (Balogh et al., 2022). Participants approach encountered problems analytically and develop various strategies, thereby demonstrating their performance in the field of information security. Furthermore, implementing CTF activities by providing tasks based on real world scenarios is an effective method for supporting experiential learning, promoting lifelong learning, and fostering cybersecurity awareness. Through CTF competitions, participants apply their existing theoretical knowledge to uncover hidden information while simultaneously testing their practical skills. Initially designed for entertainment and competition, these contests have recently been adopted by educational institutions and public organizations as pedagogical tools that enhance the learning experience (Melzer et al., 2024). In this context, CTF competitions allow participants to develop both cognitive and social skills, including problem solving, critical thinking, teamwork, and time management. The study, conducted at a technological university in Ireland, indicates that CTF competitions support Kolb's Experiential Learning Model, which posits that learning occurs through a cyclical process of experiencing, reflecting, conceptualizing and applying (Gleeson, 2024). Another significant aspect of CTF competitions is their role in raising awareness of ethical hacking. Penetration testing, also known as ethical hacking, involves a series of procedures that legally emulate the behaviors of potential attackers. In this process, identified vulnerabilities are documented and practical recommendations are provided for mitigating security gaps, rather than being exploited maliciously (Aibekova and Selvarajah, 2022). Through participation in CTF competitions, individuals are also introduced to a culture of safe and responsible computing. Chain et al., investigated how CTF competitions can be conducted on a Cloud based attack defense platform and demonstrated that the Cyber Defense Exercise (CDX) infrastructure is suitable for this purpose. Such cloud-supported

systems provide students with a scalable, multi layered, and secure laboratory environment (Chain et al., 2018).

With the acceleration of digitalization, cybersecurity has become critically important in both academic and industrial domains. CTF competitions are not limited to academic institutions; military and industrial organizations have also adopted this approach. Timmins et al., in their "Offensive Cyber Security Trainer" project focused on the Platform Management System (PMS) for the Royal Canadian Navy, demonstrated that CTF-based simulations can be utilized for cybersecurity training on naval platforms (Timmins et al., 2021). CTF competitions are generally organized into three main categories: Jeopardy, Attack-Defense, and Mixed. In the Jeopardy category, participants solve independent challenges across various domains; in the Attack-Defense category, participants both protect their own systems and attempt to gain points by attacking competitors' systems; and in the Mixed category, participants perform tasks expected in both of the other categories (Cole, 2022).

While many CTF-based studies focus on isolated challenges or competition-oriented designs, this study presents a structured, end-to-end CTF scenario based on a realistic banking system. The framework follows a sequential instructional flow that reflects a real world attack chain, from reconnaissance and initial access to privilege escalation and goal completion.

The novelty of the proposed approach lies in its multi path exploitation design, allowing participants to reach the same objective through different attack vectors, as well as in the integration of a script based monitoring mechanism that ensures traceability and instructional feedback. By aligning each stage of the scenario with specific learning objectives, the model bridges technical realism and pedagogy, offering a reusable and instruction oriented framework for applied cybersecurity education.

In light of the literature discussed above, it can be concluded that CTF competitions serve as participant-centered learning tools that facilitate the transition from theory to practice in cybersecurity education while also motivating learners. The reviewed studies further indicate that CTF environments enhance students' technical skills, increase their motivation, and strengthen defense awareness at the organizational level. Additionally, the studies demonstrate that CTF competitions are suitable for cybersecurity modules. Academic research obtained through searches in academic databases and indexes using keywords such as cybersecurity, cybersecurity education, and capture the flag is presented in Table 1.

Table 1. CTF-Based Cybersecurity Studies

No	Author/Authors	Subject/Field	Flag Capture Type	Target Audience/Education Level	Topic	Key Information
1	Nagare et al., 2025	The Role of CTF Competitions in Cybersecurity Training	Capture the Flag Scenario Pentest practice and instruction for beginners	Realistic testing environment and evaluation	CTF-Based Cybersecurity Education	Vulnerability-based CTF tasks enhanced cybersecurity knowledge.
2	Melzer et al., 2024	5G/IoT security training	Jeopardy	5G engineers, IoT security specialists, advanced students	5G protocols	Free5GC, UERANSIM and CTFd integration, packet injection-based automated scoring, 5G challenge suite
3	Gleeson, 2024	Pedagogical case study/student experience	Jeopardy	University undergraduate students	Student experience, motivation, competition	Qualitative findings: Overspecialization, need for mentorship, differences in experience, and the impact of competition on learning
4	Aibekova and Selvarajah, 2022	Penetration testing training examples and tools	Vulnhub Five86-1 in Capture the Flag Format	Pentest practice and instruction for beginners	The use of tools like Nmap, Metasploit, and Hashcat, and the PTES methodology	Specific penetration testing steps and tools used in flag capture, Vulnhub example for training purposes
5	Balogh et al., 2022	Honeypots and data collection for SOC	Capture the Flag Scenario	University students, SOC research trainings	Honeypot design, data analysis, monitoring of aggressive behavior	Data collection through honeypot and flag capture, analysis of attacker interaction time with the honeypot, and time gained for defense
6	Uddin et al., 2021	Flag capture toolkit/Basic training tools	Jeopardy	Beginners and first-level students, those new to flag capture	Presenting flag-capturing tools together accelerates the learning process, and includes an analysis of popular flag-capturing techniques and tools	Flag capture toolkit proposal, 3600 solution analysis, beginner-friendly modular challenge design
7	Sharma, 2021	Flag capture design	Jeopardy	Recruitment assessment, practical	Score calculation, integration with	Platform suggestion: Better scoring, easier recruitment

				penetration testing training	recruitment processes	
8	Timmins et al., 2021	Defense training for military platforms	Capture the Flag Scenario	Military personnel, platform operators	Kill-Chain based training and evaluation	Kill-Chain training, pedagogical design and evaluation framework
9	Singh, 2021	Software security flag capture	Code Analysis, Vulnerability Detection	Software security training for engineering students	Code security, secure software development, vulnerability research	The use of flag capture in software security education
10	Chain et al., 2018	Cloud-based attack and defense	Attack–Defense	Military/corporate training is exercise-focused	Realistic testing environment, scoring module, online monitoring and evaluation	Applications include test environment integration, network-based attack and defense, and military manpower training

2. Materials and Methods

Within the scope of this study, a simulated banking system was developed. The objective for participants is to identify deliberately embedded vulnerabilities within the system and exploit these weaknesses to carry out penetration attempts. The scenario includes security flaws such as unused open ports, weak passwords, and similar system misconfigurations. Participants are expected to detect these weaknesses and apply various attack techniques including privilege escalation, SQL injection, command injection, directory traversal, brute force, and steganography to capture the three flags. Participants achieve their final objective once they successfully infiltrate the system and complete a money transfer to their own account.

2. 1. CTF Platform Design

During the CTF design process, a web server with the specifications provided in Table 2 was rented, as running the vulnerable server directly would pose a security risk. A file and user structure was then created on the rented server.

Table 2. Server Features

Hardware/Software	Feature
Processor (CPU)	Intel i5 4 Core
Memory (RAM)	8 GB
Storage (SSD)	256 GB
Operating System	Linux (Ubuntu 22.04)

For CTF code development, Visual Studio Code (VS Code), an open source code editor developed by Microsoft, was utilized with the help of Microsoft Visual Studio Code Documentation. VS Code facilitates the software development process through features such as coding, debugging, version control (Git), and plugin support. In this study, it served as the primary development environment for creating and editing Java, PHP,

HTML, CSS, and SQL code. MySQL was selected as the database management system. The programming languages used in this study and the rationale for their selection are presented in Table 3.

Table 3. Languages Used and Reasons for Preference

Language	Reason of Usage	Reason for Preference
Java	Execution of backend operations, creation of server-side business logic.	Its secure object-oriented architecture and high performance make it suitable for large-scale backend operations.
PHP	Creating dynamic web pages and enabling interaction with a database.	Due to its widespread adoption, ease of use, and strong database integration capabilities, PHP is well suited for dynamic web application development.
SQL	For database management, including data insertion, deletion, updating, and querying operations.	SQL enables structured, efficient, and secure database management, which is essential for handling sensitive financial data.
HTML	Creating the structural framework of web pages.	HTML serves as the standard markup language for structuring web pages and ensures cross-browser compatibility.
CSS	Creating the visual design and style of web pages.	CSS enhances user experience by improving visual consistency and interface aesthetics.

This CTF environment was developed solely for educational and research purposes within a controlled and authorized setting. All offensive security techniques were applied against an intentionally vulnerable system prepared by the authors, and no real banking infrastructure, real customer data, or unauthorized targets were involved. Participants were informed about ethical hacking principles, scope boundaries, and responsible use expectations. The objective of including offensive techniques is to improve defensive awareness and secure development/operation practices, rather than enabling misuse.

2. 2. CTF Attack Techniques

To simulate the detection and exploitation of system vulnerabilities, the study employed several attack techniques, including privilege escalation, SQL injection, command injection, directory traversal, brute force, and steganography.

2. 2. 1. SQL Injection: SQL injection is an attack technique in which malicious inputs are used to manipulate database queries due to insufficient input validation, allowing unauthorized access or modification of data (Oorschot, trans. Bıçakçı, 2022).

2. 2. 2. Privilege Escalation: Privilege escalation refers to exploiting system or application vulnerabilities to gain higher access rights than originally assigned to a user (Safe Security, 2021). This technique can occur in two forms: It may occur horizontally, by accessing resources of users with similar privileges, or vertically, by obtaining higher level administrative permissions (De Pasquale et al., 2024).

2. 2. 3. Command Injection: Command injection occurs when insufficient input validation allows attackers to execute operating system commands through a web application, potentially leading to system compromise or unauthorized output retrieval (Stasinopoulos et al., 2019).

2. 2. 4. Directory Traversal: Directory traversal arises when unvalidated path inputs enable access to unintended directories and sensitive files outside the permitted scope, potentially resulting in data exposure or code execution (Flanders, 2019).

2. 2. 5. Brute Force: Brute force attacks attempt large numbers of credential combinations to bypass authentication, often exploiting weak passwords and poorly secured services (Stiawan et al., 2019). In this study, participants use tools such as Hydra and John the Ripper.

Hydra is a tool used for conducting brute force–based authentication attacks in network environments. It is particularly effective on remote access protocols such as Telnet, where it automates a large number of authentication attempts by leveraging wordlist files containing username–password pairs. This approach enables systematic password-guessing attempts against both active and passive user accounts, thereby increasing the practical feasibility of brute force attacks (Amijoyo et al., 2020).

John the Ripper is an open-source brute force tool used by both security analysts and malicious actors. The software performs particularly well in dictionary-based attacks targeting commonly used passwords; however, without additional configuration, it demonstrates more limited effectiveness in full brute force methods. In addition, the tool's extensive rainbow table support enables the rapid cracking of frequently used passwords and their variations (Marchetti and Bodily, 2022).

2. 2. 6. Steganography: Steganography is a method of concealing the existence of information by embedding it within other information, typically in image, audio, or text files. The primary objective is to ensure that even the presence of hidden communication remains undetectable. Digital images are most commonly used, and different techniques are employed depending on specific requirements (Morkel et al., 2005).

2. 3. CTF Scenario

Within the scope of this study, a bank system simulation was developed. The objective for the participants is to identify intentionally embedded vulnerabilities within the system and to attempt to infiltrate it by exploiting these weaknesses. The scenario includes issues such as unused open ports, security flaws related to hidden files/passwords, and similar vulnerabilities. Participants are expected to detect these weaknesses and apply attack techniques including privilege escalation, SQL injection, command injection, directory traversal, brute-force attacks, and steganography, ultimately capturing the three flags. Participants achieve their goal once they successfully compromise the system and perform a money transfer to their own accounts. The flowchart of the implemented study is presented in Figure 2.

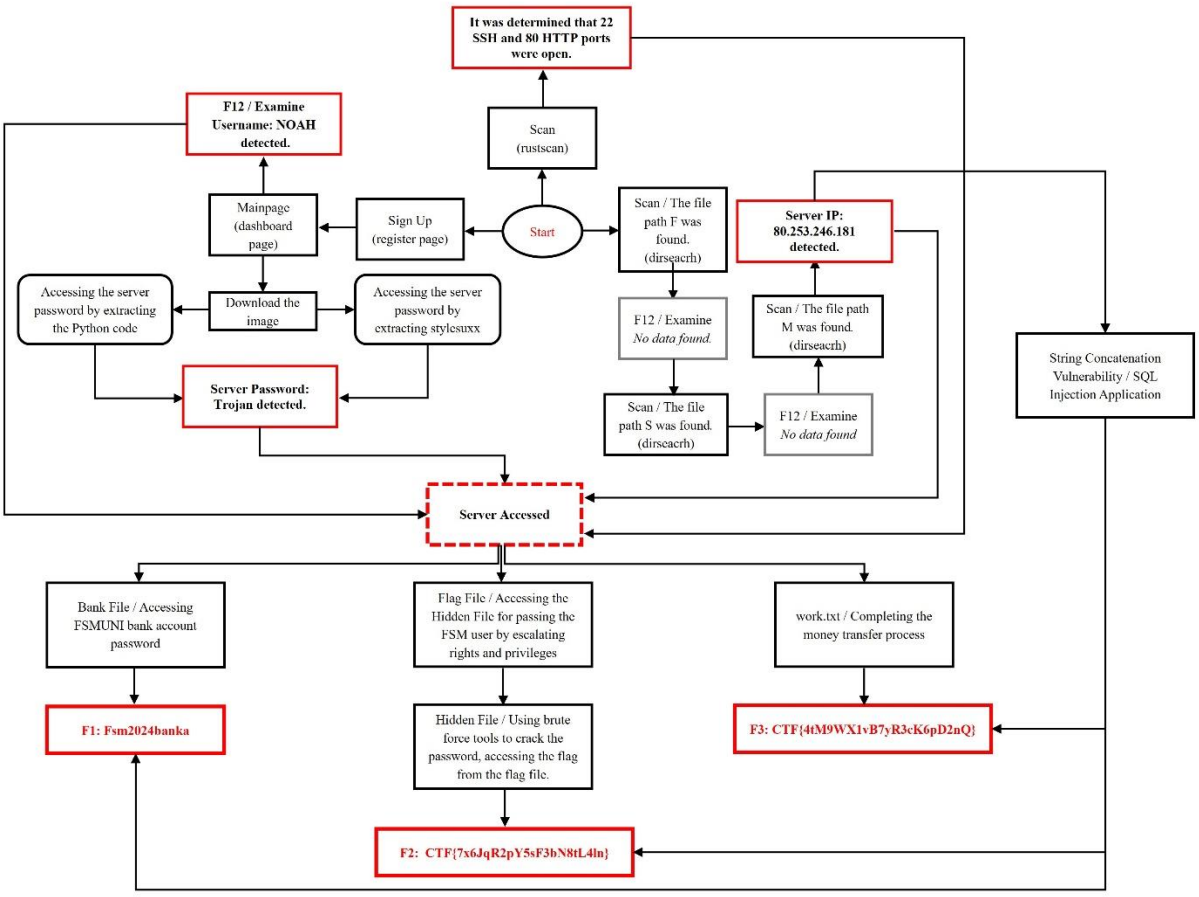

Figure 2. CTF Flowchart

In Figure 2, different colors are used to visually distinguish the functional stages of the CTF workflow. In Figure 2, different frame styles and colors are used to clarify the functional roles of each stage within the CTF workflow. Stages related to reconnaissance and initial system access are represented with black frames. Target oriented stages, including flag acquisition and completion of the simulated banking task, are indicated with red frames. Tasks required to reach the server are shown with black text and red outer frames, highlighting their intermediary role in the attack flow.

Access to the server is further emphasized using red dashed frames, while misleading or decoy elements are illustrated with gray frames. In addition, alternative paths that can be used to obtain the server password are depicted using rounded rectangles, indicating multiple valid solution strategies within the scenario. These visual distinctions are intended to improve readability and enable accurate interpretation of the sequential and multi path structure of the proposed CTF scenario.

In the banking simulation, participants are first greeted by the login page. At this stage, a three step game flow presented in Figure 3 guides participants toward reaching the flags. This workflow includes registering for the banking system, scanning open ports using the Rustscan tool, and accessing hidden files and directories placed within the target web application via the Dirsearch tool. Rustscan operates efficiently and rapidly in identifying services running on open ports of a device. It is one of the publicly accessible security testing tools used for automated port identification and facilitates the concurrent execution of numerous tasks on the operating system. It is commonly employed in penetration testing,

network discovery, and vulnerability analysis (Temiz, 2022; Taupaani and Harwahyu, 2025). Dirsearch, on the other hand, is a Python-based command-line tool. Designed for attacking directories and files on web servers, it can run on Windows, Linux, and macOS.

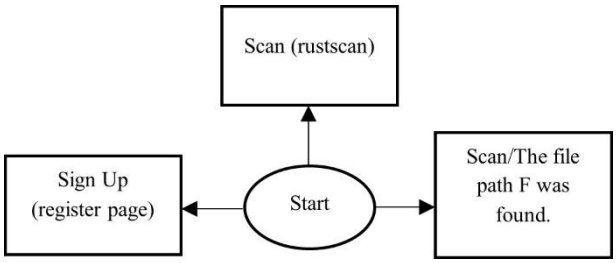

Figure 3. CTF Starting Points

By analyzing HTTP response codes, redirects, and server behaviors, the tool enables the discovery of potentially concealed files and directories. It is frequently used in penetration testing, Red Team operations, and web security assessments (Utama and Nurhadi, 2024). The workflow of these methods is provided below.

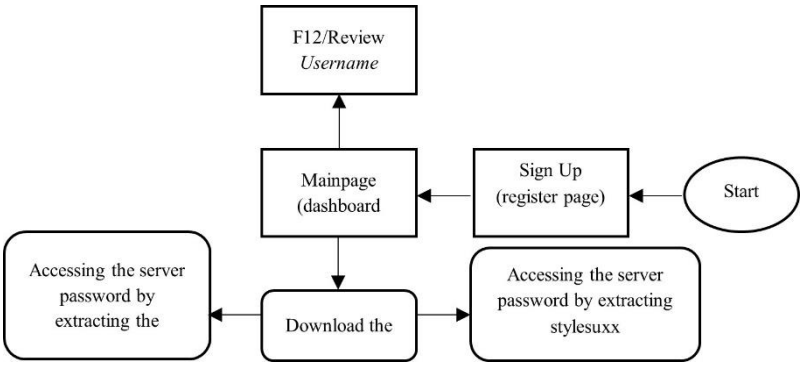

Figure 4. Obtaining Username / Password in a CTF Environment

1) When the participant scans the login page using the Rustscan tool, ports 22 (SSH) and 80 (HTTP) are identified as open.
2) When a scan is performed with the Dirsearch tool, the F file path can be reached. However, examining the page using F12 does not provide access to the data. The second Dirsearch scan reveals the S file path, though F12 inspection once again does not grant access to the data. The third Dirsearch scan uncovers the M file path, and inspecting it with F12 reveals the server IP address: 80.253.246.181. From this point forward, the CTF can be solved using two different methods.

Table 4. Python Function for Extracting Server Password

```
from PIL import Image
from PIL.ExifTags import TAGS
# A simple function that extracts EXIF information
def get_exif_info(file_path):
    img = Image.open(file_path)
    exif_data = img._getexif()
    if not exif_data:
        print("This image does not contain EXIF
information.")
        return
    for label, value in exif_data.items():
        label_name = TAGS.get(label, label)
        print(f"{label_name}: {value}")
# Example usage
get_exif_info ("b.png")
```

2.1. When the participant navigates to the registration page, they register for the banking system using their user information and are subsequently redirected to the homepage. Two different game flows awaiting the participant at this stage of the game are presented in Figure 4.

a) By opening the developer tools via F12, the participant can obtain the username NOAH.

b) When the participant downloads the image located on the homepage, the server password can be obtained using one of the two methods below:

i. By using the Python function presented in Table 4, the necessary data are extracted, enabling access to the server password of Truva.

ii. The server password specifically the Truva credential can also be obtained by extracting hidden text using the browser-based Stylesuxx webpage, which is designed for revealing concealed content.

As a result of all these steps, the information required to connect to the server via SSH namely port 22, the username NOAH@80.253.246.181, and the password Truva can be obtained. After gaining access to the server, the objective is to locate three different flags embedded within the system. The requirements and contents related to these flags are presented in Table 5.

Table 5. Flags

Flags	Requirement	Flag Content / Explanation
F1: Fsm2024banka	Accessing to bank file	FSMUNI bank account password
F2: CTF{7x6JqR2pY5sF3bN8tL4ln}	Accessing to the hidden file and *wordlist.txt*; passing to the FSM user by escalating privileges	The code is located in the file named *flag*
F3: CTF{4tM9WX1vB7yR3cK6pD2nQ}	Completing the money transfer process as described in the *work.txt* file	The code displayed on the screen when the process is complete

Within the server, the FSMUNI bank account password is located inside the bank directory, and this password provides the first flag. When the participant locates the file named “Flag” they will encounter an access restriction. At this point, the participant is expected to escalate privileges and switch to the FSM user. To obtain this privilege, once the hidden file and the wordlist.txt file within the directories are accessed, the password of the hidden file must be cracked using brute-force tools such as Hydra or John the Ripper. After obtaining the file password—Myharley—the participant can switch to the FSM user and retrieve the second flag from the file named “Flag”.

After completing the money transfer process described in work.txt, the participant can also obtain the third flag.

2.2. After obtaining the IP address from the F, S, and M directories, the participant can pursue an alternative solution path by analyzing the interaction between the web application and its database. In the test environment, a dynamically concatenated query structure generally not recommended for security purposes was intentionally used instead of a parameterized query structure in order to raise participants’ vulnerability awareness. Incorporating user input directly into the query without any parameter checks or input validation mechanisms results in a vulnerable database interaction. This vulnerability, known as SQL Injection enabled through string concatenation, is triggered via the 'username' input field on the login page. To illustrate the situation, Table 6 presents the vulnerable query structure.

Table 6. Vulnerable Database Query Structure

```
username = input("Enter your username: ")
query = "SELECT * FROM users WHERE username
= '" + username + "'"
print("Created query:")
print(query)
```

This vulnerable structure enables the participant to execute unauthorized queries on the database. For example, using an input such as 'admin' OR '1'='1', the resulting query becomes SELECT * FROM users WHERE username = 'admin' OR '1'='1'. This query always returns a true condition. Consequently, access to the database can be achieved without any user authentication, allowing the participant to run unauthorized queries and obtain the three flags hidden within the scenario. The pseudocode flow related to this process is presented in Table 7.

Table 7. Pseudocode Flow for Detecting SQL Injection Vulnerabilities

```
START
Enter test data into the user input field
IF the application behaves abnormally as follows:
  - If it produces an unusual error message
  - If it makes an unexpected redirection
  - If the response time changes
THEN
  SQL injection is possible
The participant follows these steps:
1. Attempts normal input
2. Then attempts a test input containing special
control characters
3. Compares server responses
IF unexpected data is returned from the database
THEN
  SQL injection vulnerability is confirmed
  Data leading to the flag is obtained
END
```

When the participant submits the captured flag codes on the relevant page, feedback is provided regarding the correctness of the codes. In cases of incorrect submissions, the participant is notified and allowed to return to the corresponding stages of the game. Additionally, a script developed to track the commands executed by all users in the terminal is presented in Table 8. Through this mechanism, it becomes possible to observe in detail how close each user is to obtaining the flags, who has completed the process, and which paths they followed. When all three flags are correctly entered on the designated page, the CTF is successfully completed.

Table 8. Tracking Script

```
import os
import time
log_file = "/var/log/ctf_activity.log"
while True:
   os.system("history 1 >> " + log_file)
   time.sleep(2)
```

3. Results and Discussions

The Capture the Flag (CTF) application, designed around banking operations and aimed at having participants discover intentionally embedded security vulnerabilities and exploit them to attempt system intrusions, provides participants with an opportunity to experience realistic attack and defense processes. Throughout the application process, participants' progress was monitored using a tracking script to assess the effectiveness of vulnerabilities such as command injection, SQL injection, directory traversal, and steganography. Moreover, based on the findings obtained through the script, the exploration of

server access, privilege escalation, and brüte force vulnerabilities was tracked. Literature reviews indicate that the high prevalence of privilege escalation attacks underscores the necessity of enhancing educational capacity regarding exploitation techniques aimed at unauthorized control of system resources (Bayazit and Arac, 2025).

The staged tasks within the CTF were designed to be challenging for participants. Sequential logic-based steps, such as accessing hidden directories, required the combined use of network security, scripting, file permissions, and system analysis. Progressing to server access in the initial stage allowed participants to gain valuable experience in solving layered security problems. The structured, stepwise implementation of CTF solution processes provided participants with an in depth understanding of the underlying logic of cyberattack procedures. The process was particularly beneficial for developing a systematic approach to vulnerability analysis, interpreting attack steps within a cause and effect framework, and understanding the operational impact of applied techniques. From an instructional perspective, the monitored progress indicates that participants achieve better learning outcomes when tasks require the integration of multiple competencies, such as web enumeration, credential acquisition, and secure shell access, rather than the isolated application of a single technique. The repeated attempts observed in more advanced stages, particularly during privilege escalation and database driven exploitation, suggest that these steps act as critical learning thresholds. At these points, additional instructional support such as guided exercises or structured hints can substantially enhance skill acquisition. Accordingly, the proposed scenario not only replicates a realistic attack chain but also offers actionable insights into where learners most commonly require targeted educational support.

3.1. Qualitative Evaluation and Observational Insights

Although this study does not report quantitative performance metrics or statistical analyses, participant interactions with the CTF environment were continuously observed through the script based monitoring mechanism and flag submission process. These observations enabled an evaluation of how participants progressed through the sequential stages of the scenario and which tasks required repeated attempts or additional exploration. This limitation is inherent to the exploratory and practice oriented nature of the proposed CTF based educational framework, which primarily emphasizes instructional design, scenario realism, and qualitative learning outcomes rather than measurable performance statistics. Nevertheless, future work may incorporate quantitative evaluation indicators such as task completion rates, time-to-flag measurements, error frequencies, and time based learning progression analyses to complement the qualitative observations and enable a more comprehensive assessment of learner performance.

The findings indicate that participants generally advanced smoothly through the initial reconnaissance and access phases, while later stages particularly those involving privilege escalation and chained exploitation required more iterative problem solving. This pattern suggests that multi stage CTF scenarios effectively reveal learning thresholds where additional instructional support or scaffolding may be beneficial. From an educational perspective, these observations support the role of structured

CTF environments as diagnostic tools for identifying both technical strengths and learning gaps in applied cybersecurity training.

This study, based on the CTF methodology commonly used in penetration testing in the cybersecurity sector demonstrated that challenges encountered during database based attacks and privilege escalation exercises highlight the need to strengthen practical experience in these areas. This underscores the necessity of deepening educational content on relevant technical topics and increasing the use of application oriented teaching strategies. The findings indicate that CTF applications support participants in understanding the flow of attack processes and in developing a systematic approach to vulnerability assessment. At the same time, difficulties encountered in advanced techniques such as SQL injection and privilege escalation suggest a need for additional practical exercises and instructional materials in these areas. Overall, the results reveal that while CTF scenarios are effective in enhancing technical competency, enriching the educational content in terms of scope and depth has a significant positive impact on learning outcomes.

4. Conclusions

This study presents a structured, CTF based cybersecurity training model built around a realistic banking system simulation. By integrating multiple attack techniques into a sequential and reproducible scenario, the proposed framework enables learners to experience end-to-end penetration testing processes in a controlled educational setting. The results indicate that such structured CTF environments effectively support the transition from theoretical knowledge to practical skill development while revealing critical learning stages that benefit from guided instructional support. Overall, the study demonstrates that scenario driven CTF designs can serve as effective and reusable tools for applied cybersecurity education and awareness. In conclusion, the developed banking system simulation provided a comprehensive practical environment for applying penetration-testing tools and techniques while significantly enhancing participants' awareness of identifying and mitigating security vulnerabilities.

Conflict of Interest

The authors stated that there are no conflicts of interest regarding the publication of this article.

Researchers' Contribution Rate Statement

The contribution rates of the authors in the study are equal.

Referances

Aibekova A., Selvarajah V. Offensive security: study on penetration testing attacks, methods, and their types. 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 23-24 April 2022; 1-9, Ballari, India.

Amijoyo T., Umar R., Yudhana A. Bruteforce in the hydra process and telnet service using the naïve bayes method. Jurnal Mantik 2020; 4(1): 319-326.

Balogh Á., Érsok M., Erdődi L., Szarvák A., Kail E., Bánáti A. Honeypot optimization based on CTF game. 2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI), 02-05 March 2022; 153-158, Poprad, Slovakia.

Bayazit EC., Arac M. Implementing a method for privilege escalation attacks in windows systems. 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA), 23-24 May 2025; 1-7, Ankara, Turkiye.

Chain K., Kuo CC., Liu IH., Li JS., Yang CS. Design and implement of capture the flag based on cloud offense and defense platform. 2018 IEEE International Conference on Applied System Invention (ICASI), 3-17 April 2018; 686-689, Chiba, Japan.

Cole SV. Impact of capture the flag (CTF)-style vs. traditional exercises in an ıntroductory computer security class. 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 1 (ITiCSE 2022), 07 July 2022; 470-476, Dublin, Ireland.

De Pasquale G., Grishchenko I., Iesari R., Pizarro G., Cavallaro L., Kruegel C., Vigna G. ChainReactor: Automated privilege escalation chain discovery via AI planning. 33rd USENIX Security Symp, 12 August 2024; 5913–5929, Philadelphia, PA, USA.

Flanders M. A simple and intuitive algorithm for preventing directory traversal attacks. CoRR abs/1908.04502, 13 August 2019; arXiv:1908.04502.

Gardiner J., Abaimov S., Williams J., Shahbi F., Anastasakis K., Chowdhury PD., Ellis W., Sameen M., Samanis E., Rashid A. If you build it, they will come — a blueprint for ıcs-focused capture-the-flag competitions. In Proceedings of the Sixth Workshop on CPS & IoT Security and Privacy (CPSIoTSec 2024), 22 November 2024; 27–40, Salt Lake City, Utah, USA.

Gleeson M. Cybersecurity students'experiences of capture the flag (Ctf) in an irish technological university. In 2024 Cyber Research Conference (Cyber-RCI). 25-25 November 2024; 1–9, Carlow, Ireland.

International Telecommunication Union (ITU), Global cybersecurity index 2024 report. https://www.itu.int/en/ITU-D/Cybersecurity/pages/global-cybersecurity-index.aspx (accessed at 02 December 2025)

ISACA, State of Cybersecurity 2024. https://www.isaca.org/resources/reports/state-of-cybersecurity-2024 (accessed at 20 December 2025)

Marchetti K., Bodily P. John the ripper: an examination and analysis of the popular hash cracking algorithm. 2022 Intermountain Engineering, Technology and Computing (IETC); 13-14 May 2022;1-6, Orem, UT, USA.

Melzer J., Shafo A., Zhao Z., Wang PP., Fenwick W., Almuhtadi W. Developing "capture the flag" for 5g ıot cyber security training. 2024 IEEE 10th World Forum on Internet of Things (WF-IoT), 10-13 November 2024; 816-821, Ottawa, Canada.

Morkel T., Eloff JHP., Olivier MS. An overview of image steganography. 5th Annu. Inf. Secur. South Africa Conference, June 2005; 1–11, Pretoria, South Africa.

Nagare Y., Saini JS., Parasiya D., Kumar V., Jain N., Sikdar P. From simulation to application: The role of CTF competitions in cybersecurity training. 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 06-08 March 2025; 1-6, Gwalior, India.

Oorschot P. Siber güvenliğe Giriş (K. Bıçakçı, Çev.). 1th ed. Palme Press; 2022.

Safe Security, A hands-on approach to Linux privilege escalation. https://safe.security/wp-content/uploads/a-hands-on-approach-to-linux-privilege-escalation.pdf (accessed at 28 November 2025)

Sharma S., Kumar R., Jain A., Agarwal B., Suman SK. Intensifying practical based learning of penetration testing using CTF. 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). 17-18 December 2021; 378-1381, Greater Noida, India.

Singh R. Software security (capture the flag). 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT). July 2021; 165-168, Sonepat, India.

Stasinopoulos A., Ntantogian C., Xenakis C. Commix: automating evaluation and exploitation of command injection vulnerabilities in web applications. International Journal of Information Security 2019; 18(1): 49-72.

Stiawan D., Idris MY., Malik RF., Nurmaini S., Alsharif N., Budiarto R. Investigating brute force attack patterns in IoT network. Journal of Electrical and Computer Engineering 2019; 1: 4568368

Taupaani R., Harwahyu R. ZTSCAN: Enhancing zero trust resource discovery with masscan and NMAP integratıon. Jurnal Ilmu Pengetahuan dan Teknologi Komputer 2025; 10(4): 868-877.

Temiz H., Büyükeke A. An ınverse approach to windows' resource-based permission mechanism for access permission vulnerability detection. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2022; 5(2): 534-550.

Timmins J., Knight S., Lachine B. Offensive cyber security trainer for platform management systems. 2021 IEEE International Systems Conference (SysCon), 15 April 2021; 1-8, Vancouver BC, Canada.

Uddin MN., Rastogi T., Mishra S., Verma A., Das A., Kothari P. An implementation of capture the flag (CTF) tool. 2021 International Conference on Technological Advancements and Innovations (ICTAI), November 2021; 88-393, Tashkent, Uzbekistan.

Utama FP., Nurhadi RM. Uncovering the risk of academic information system vulnerability through PTES and OWASP method. Communication and Information Technology Journal 2024; 18(1): 39-51.