Improved Method for Penetration Testing of Web Applications

Aasso Ziro[a,b], Sergiy Gnatyuk[c] and Shara Toibayeva[d]

[a] *Al-Farabi Kazakh National University, 71 al-Farabi Ave., Almaty, 050040, Republic of Kazakhstan*
[b] *Kazakh-British Technical University, Tole bi street, 59, Almaty, 050000, Republic of Kazakhstan*
[c] *National Aviation University, Kosmonavta Komarova avenue, 1, Kyiv, 05038, Ukraine*
[d] *University of Power Engineering and Telecommunications (AUPET) named after G. Daukeev, Baytursynuli 126/1, Almaty, 050013, Republic of Kazakhstan*

Abstract

Penetration testing is a crucial aspect of information security and is used to access the security posture of a system. The objective of this research is to analyze the various methods and techniques used in penetration testing. The study focuses on the different stages of penetration testing, including planning, reconnaissance, scanning, exploiting, and reporting. A comprehensive analysis of the commonly used tools and techniques in each stage is provided, along with a discussion of their strengths and weaknesses. The research also provides a critical evaluation of the current state of the art in penetration testing and identifies areas for improvement. The study concludes by presenting recommendations for future research in this field. The results of this research are valuable for organizations and individuals who want to enhance their security posture and protect against potential threats.

Keywords 1

Audit, information security, penetration testing, SecOPs, IDS, code analysis, Kali Linux.

1. Introduction

The purpose of the research is to provide a comprehensive understanding of the methods and techniques used in penetration testing, with a focus on identifying best practices and areas for improvement. The goal of this research is to help organizations optimize their information security posture by providing valuable insights into the planning, reconnaissance, scanning, exploiting, and reporting phases of penetration testing. By conducting a thorough analysis of the existing literature on penetration testing methodology, the research aims to provide organizations with a valuable resource for improving their information security assessment process and protecting their sensitive information against potential threats. Ultimately, the purpose of this research is to contribute to the ongoing efforts to enhance the effectiveness and efficiency of penetration testing as a tool for information security assessment [1].

The purpose of this research is to analyze the current state of penetration testing methodology, identify best practices and challenges, and suggest improvements for the future. The research will focus on the following areas:

1. Definition and scope of penetration testing: The research will examine the definition of penetration testing and its scope, including the types of tests that are commonly performed, the objectives, and the types of systems and applications that are typically tested.
2. Current state of penetration testing: The research will review the current state of penetration testing, including the tools and techniques that are used, the approaches and methodologies that are followed, and the results that are generated.

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: ziro.aasso@gmail.com (A. Ziro); sergio.gnatyuk@gmail.com (S. Gnatyuk); shara_t@mail.ru (Sh.Toibayeva)
ORCID: 0000-0002-5952-877X (A. Ziro); 0000-0003-4992-0564 (S. Gnatyuk); 0000-0002-2027-0396 (Sh.Toibayeva)

3. Best practices and challenges: The research will examine the best practices and challenges that organizations face in conducting penetration testing. This will include the use of frameworks and standards, the selection of testing tools, the design of test cases, and the interpretation of results.
4. Future improvements: The research will suggest improvements for the future of penetration testing, including the development of new tools and techniques, the adoption of standards and frameworks, the integration of penetration testing with other security practices, and the use of advanced analytics and machine learning.

2. Related papers analysis

During scientific research, it was found that when testing for penetration, it is necessary to know the basic methods of attacks on the information system, as well as having the skills of ethical hacking, it will help to compile a model of a neural network where one or another technology is temporarily selected.

The article "Offensive Security: A Study of Attacks, methods and their types for penetration testing" provides an overview of the concept of penetration testing, also known as ethical hacking. This underscores the importance of this process in the era of digitalization, when the development of technology has brought many useful and effective services into people's lives, but also opened the door for attackers who use information to steal valuable data [2]. The article explains that penetration testing is a set of procedures that simulate the actions of potential hackers to identify weaknesses in the system. The purpose of the study is to discuss the different types of penetration testing, the strategies used, the code of conduct for penetration testers and the methodology used. The article describes a practical exercise in the form of CTF format using the Five 86-1 machine and the Kali Linux operating system to demonstrate various attacks. The article also discusses the consequences and critical analysis of these attacks.

It should be noted that in the article «Off-the-Shelf Solutions as Potential Cyber Threats to Industrial Environments and Simple-To-Implement Protection Methodology» focuses on exploring cyber threats and potential solutions for protecting industrial control systems (ICS) [3]. It examines different ready-made offensive solutions, including hardware and software, and tests their effectiveness as attack vectors to gain access to usually unprotected industrial installations. The article demonstrates man-in-the-middle (MITM) and Legal client-server (LCSA) attacks using the Modbus communication protocol implemented at a real compressor station. In order to raise awareness about cyber threats, the article showcases how these ready-made solutions can be used in real-life attacks. The article also suggests a simple and cost-effective hardening technique that can be applied in industrial enterprises to protect against cyber threats. Additionally, a new method of protection based on PLC is discussed and demonstrated, which consists of a mechanism for monitoring signal reliability and verifying the integrity of the control system. The experimental results of both penetration testing and the hardening technique are tested using real PLCs and HMI devices.

During the research, the article "Automated Penetration Testing based on a threat model" describes a study in which an algorithm for systematic penetration testing based on a threat model is proposed [4]. The purpose of the study is to make sure that all existing threats to the system are checked and considered in the penetration testing process. The research focuses on assembling a package of penetration testing tools and following standard methodologies for developing automated penetration testing. The study uses a threat model developed at the IT Innovation Center as a starting point for penetration testing and complies with the NIST 800-115 standard for penetration testing. The ultimate goal of the study is to reduce the consequences of malicious attacks by applying the proposed algorithm of automated penetration testing to a real system.

During the analysis of the articles, it was revealed that penetration testing in the modern world plays an important role in protecting the organization's information.

3. Problem Statement

The research will be conducted using a combination of qualitative and quantitative methods, including literature review, surveys, and case studies. The literature review will focus on existing

studies and reports on penetration testing, while the surveys will gather data from security experts, practitioners, and organizations that have conducted penetration testing. The case studies will provide detailed insights into the penetration testing process and the results that were generated [5].

The results of this research will be valuable to a wide range of stakeholders, including security experts, practitioners, organizations, and academic researchers. For security experts and practitioners, the research will provide a comprehensive overview of the current state of penetration testing, best practices, and challenges, and suggest ways to improve the process. For organizations, the research will provide valuable insights into the security posture of their information systems and help to identify areas for improvement. For academic researchers, the research will provide a foundation for future research in penetration testing.

4. Proposed method

White box penetration testing is a methodology that involves in-depth knowledge of the target system's architecture, configuration, and source code. It is a comprehensive approach that includes both the internal and external perspectives of the target system and provides a more in-depth evaluation of the system's security posture [6].

In contrast to black box testing, which only focuses on the external perspective of the target system and does not involve any prior knowledge of the system, white box testing provides a much more thorough examination of the system's security. This is because the tester has access to the source code and configuration information of the system, which enables them to identify potential security vulnerabilities that would otherwise be missed by black box testing. The objective of white box testing is to evaluate the security of a target system from an internal perspective. This involves the examination of the system's architecture, design, and implementation to identify potential security vulnerabilities that may arise from coding errors, misconfigurations, or design flaws. The tester will also evaluate the system's security controls, such as authentication, authorization, and access controls, to ensure that they are properly implemented and configured to prevent unauthorized access to sensitive information. One of the main benefits of white box testing is the ability to identify security vulnerabilities that are unique to the target system [7]. These vulnerabilities may not be discovered through black box testing or other security assessments, as they are often hidden in the source code or configuration of the system. By conducting a white box test, organizations can gain a deeper understanding of the security risks inherent in their systems and implement the necessary controls to mitigate these risks. The white box testing process typically begins with a review of the system's architecture and design, including the source code and configuration information. The tester will then perform a thorough examination of the code to identify any potential security vulnerabilities that may exist, such as buffer overflows, SQL injection, or cross-site scripting. The tester will also evaluate the system's security controls to ensure that they are properly implemented and configured to prevent unauthorized access to sensitive information.

In addition to identifying potential security vulnerabilities, white box testing can also help organizations to improve their software development processes. This is because the tester can provide feedback on the code and suggest improvements that can be made to reduce the risk of security vulnerabilities in future releases. This feedback can be valuable in improving the quality and security of the code and reducing the risk of security incidents in the future[8].

Another important benefit of white box testing is the ability to prioritize the fixing of security vulnerabilities. With black box testing, the tester can only identify potential security vulnerabilities, but cannot assess their severity. With white box testing, the tester has access to the source code and configuration information, which enables them to determine the severity of each vulnerability and prioritize the fixing of the most critical vulnerabilities.

Despite its benefits, white box testing also has some challenges. One of the main challenges is the time and resources required to conduct a comprehensive white box test. This is because the tester needs to have in-depth knowledge of the system's architecture, design, and implementation, as well as access to the source code and configuration information. Additionally, the process of reviewing the source code and identifying potential security vulnerabilities can be time-consuming and requires specialized skills and knowledge [9].

Another challenge is the potential impact on the development process. In some cases, white box testing can cause delays in the development process as the tester may need to review large amounts of code to identify potential security vulnerabilities. This can also impact the delivery of the product or service, as the fixing of security vulnerabilities may require changes to the code and configuration.
In conclusion, white box penetration testing is a comprehensive approach that provides organizations with a deeper understanding of the security risks inherent in their systems. It enables organizations to identify potential security vulnerabilities that may not be discovered through black box testing and to prioritize the fixing of the most critical vulnerabilities. This can help organizations to reduce the risk of security incidents and improve the overall security posture of their systems [10].

However, white box testing also has some challenges that need to be considered. It requires specialized skills and knowledge, as well as a significant investment of time and resources. In addition, it can impact the development process and delay the delivery of the product or service. Given these challenges, organizations need to carefully consider their security testing needs and determine whether white box testing is the appropriate methodology for their specific requirements. For organizations that have sensitive information or critical systems, white box testing may be the best option to ensure a thorough evaluation of their security posture. However, for organizations with less critical systems, black box testing may be sufficient to meet their security testing needs [11].

To effectively implement white box testing, organizations need to invest in the necessary resources, including trained and experienced testers, and have a clear understanding of the testing process and objectives. They should also develop a comprehensive testing plan that outlines the scope, objectives, and approach of the testing, and have a clear understanding of the resources required to conduct the testing.

In the SecOps methodology, white box testing is typically used in the testing phase of the security operations process. During this phase, the security team works with the penetration testing team to identify vulnerabilities in the target system and develop strategies for mitigating them.

The following are the general steps for conducting a white box penetration test using the SecOps methodology:

Planning: The penetration testing team works with the security team to identify the scope of the test, including the target system, the testing methodologies, and the expected outcomes.

Reconnaissance: The penetration testing team gathers information about the target system, including network topology, system architecture, and available services.

Vulnerability analysis: The penetration testing team analyzes the information gathered during the reconnaissance phase to identify vulnerabilities in the target system.

Exploitation: The penetration testing team attempts to exploit the identified vulnerabilities in the target system to gain access to sensitive information or to take control of the system.

Reporting: The penetration testing team reports the findings to the security team, including the identified vulnerabilities and recommended strategies for mitigating them.

Remediation: The security team uses the information provided by the penetration testing team to develop strategies for mitigating the identified vulnerabilities and improving the overall security of the target system.

Verification: The penetration testing team performs additional tests to verify that the identified vulnerabilities have been successfully mitigated and that the overall security of the target system has been improved.

By using the white box method in conjunction with the SecOps methodology, organizations can identify and mitigate security vulnerabilities in their systems, thereby reducing the risk of data breaches and other cyber attacks.

5. Experimental study

Within the framework of this study, the methodology for conducting penetration testing of a web application is described, which includes: the stages of penetration testing, the software tools used, reporting on the results of testing.

Stage № 1 – Conclusion of the contract and preparation of the terms of reference.

First of all, before conducting all testing, a contract and a technical specification must be drawn up between the company that provides testing services and the Customer, which should clearly describe the needs of the Customer: scope of work (web resources that will be tested), applicable and prohibited testing techniques (most often social engineering is prohibited, violation of the operability of a web resource and physical access to the Customer's system), requirements for reporting on test results, testing methodology, standards used [12].

Stage № 2 – Collecting information.

After the conclusion of the contract and the approval of all the necessary documents, the specialists who will directly conduct the penetration testing themselves enter business. When conducting penetration testing, one of the main stages is the collection of information on the target, which includes both passive methods of collecting information and active:

- subdomains – if according to the terms of reference, the target domain is specified as *.example.kz, then the first step is to collect all the available subdomains. Since the main domain and subdomains can be located on the same server, an attacker can break through one domain and try to gain access to other domains. The following tools can be used to collect subdomains: theHarvester, knockpy, gobuster, dnsmap. [13].

- target scanning – in addition to manual information collection, there are many automated checks (scanners) that allow you to collect a lot of information on the target in the shortest possible time, including identifying some vulnerabilities. Some scanners are aimed at specific CMS (wpscan, joomscan, drupwn), some are aimed at specific programming languages, some of the scanners are larger and check almost any system, an example of such a scanner can be Acunetix. The most commonly used is the nmap tool, which allows you to scan a target for open ports and software that runs on open ports. nmap also supports NSE scripts that can perform many automated checks, including identifying vulnerabilities.

- directories and files – the next step is to sort through the directories and files that exist in the web application. Often administrators can leave files or directories for public access that contain critical data that can be used in compound attacks or directly lead to hacking of a web resource. Such files and folders include: system directories and files in the public domain, configuration files, backups, files with logins and passwords, administrative functionality, etc. You can search for files and directories by using the following tools: feroxbuster, gobuster, dirb, BurpSuite, dirsearch, etc. As a rule, large dictionaries are used to search for such information, which can be found on the Internet, but the most experienced specialists use their own dictionaries, which were collected from various sources, including their own experience [14].

- collection of information from open sources – open sources are web resources that may contain public information about the goal, while interaction with the goal itself is not performed. Such sources include GitHub, which may contain the entire source code of the web application, forgotten by the administrator at the development stage. Such code often contains logins and passwords from databases, FTP servers, etc. Such a finding will allow a specialist to switch from Back Box testing to White Box testing and analyze the source code for vulnerabilities more efficiently, which increases the chances of finding more critical vulnerabilities [15], [16]. Also, a public tool for finding information about a goal includes Dorks (Dorks) – this is the slang name of concretized search queries, with the help of which a more thorough search is provided through the Google search engine. In addition to the above tools, there are tools that automatically know how to search for drains in leaked databases containing the login and passwords of users of the target system.

Step №3 – Search for vulnerabilities by manual method.

- fuzzing – having collected all of the above data, the specialist proceeds to direct interaction with the web resource, which implies the study of the findings, functionality and the search for vulnerabilities in it through the fuzzing technique. Fuzzing is a web application testing technique that involves transmitting specially generated data to the application input that can cause abnormal behavior in a web application. Specially generated data means so-called payloads or Payloads. Which payloads to transmit, the specialist begins to understand in the process of gaining experience in testing web applications, different types of payloads can cause different abnormal behavior, depending on many factors: the programming language in which the web resource is written, the database used, etc. For example, single and double quotation marks are used in the syntax of database queries, if the developer has not taken care of filtering the data coming from the user, the specialist may try to pass

the quotation mark and detect abnormal behavior, which, together with the payload type, gives the specialist an understanding of the possibility of SQL injection [17], [18].

Stage №4 – Post exploitation and privilege escalation.

- post exploitation and privilege escalation – in case of finding a critical vulnerability and being able to perform remote code execution on a vulnerable system, the specialist proceeds to the stage of increasing his own privileges on the server. If he manages to raise his privileges to the Administrator or root level, then a complete compromise of the system will be made, having achieved this level of privileges, the attacker can perform almost any manipulations [19], [20].

Stage №. 5 – Reporting.

- report – after conducting penetration testing, a specialist draws up a report on the vulnerabilities found, which should contain: a full description of the stage of operation of each of the vulnerabilities found, full recommendations for eliminating each of the vulnerabilities found, a gradation of vulnerabilities according to OWASP Top 10 or another methodology with a description, links to additional materials with a description of the vulnerability and recommendations for its elimination from reliable sources, a description of the actions performed by the specialist and the name of the tools used, information on the protective means that the specialist encountered if they did not allow the attack to be carried out [21].

After conducting penetration testing, an auditor enters the case whose goal is to detect all inconsistencies, including weaknesses in the internal processes of the company based on the results of penetration testing. Such an example could be a critical SQL injection vulnerability that was found during penetration testing, for an auditor, the presence of this type of vulnerabilities indicates a lack of competence in the field of information security among the developers of a web resource. All company personnel on a regular basis should take courses on awareness in the field of information security, refresher courses, cyber hygiene courses, etc. Developers should take regular refresher courses in various areas, including in the direction of information security, which improves the skills of developers, their awareness and reduces the chances of writing vulnerable web code.applications. However, the appearance of this vulnerability may indicate a number of non-working processes: the procedure for configuring the web server suffers, the installed Firewall is configured incorrectly, because of this, the payload was not noticed during testing, the SIEM system does not work correctly, which led to an unnoticeable attack, etc. As a result, the auditor is able to identify a lot of inconsistencies in the internal processes of the organization based only on the test results report.

The next step is to consider a small example based on a report on a vulnerability found during testing. As a result of testing by the White Box method, a high-level vulnerability was discovered that allows you to capture the administrative account of a web resource:

Account capture via unsafe Deserialization of an object + Type Juggling

OWASP Top 10 Classification: A2:2017 - Broken Authentication

Criticality Level: High

Vulnerable source code: /web-serveur/ch28/index.php

Vulnerable lines: 15-30

Description:

On this path web-serveur an authorization form was found in the administrative panel of the web resource, when analyzing the source code, many vulnerabilities were identified that allow an attacker to hijack the account of the administrator of the web resource. Figure 1 shows the authorization form, Figure 2 shows the authorization request.

Figure 1: Authorization form

Request

Raw | Params | Headers | Hex

```
POST /web-serveur/ch28/index.php HTTP/1.1
Host: challenge01.root-me.org
Content-Length: 26
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge01.root-me.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-e
xchange;v=b3;q=0.9
Referer: http://challenge01.root-me.org/web-serveur/ch28/
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: PHPSESSID=01c35debc56c82b6d1b7f62cbbf76d5a
Connection: close

login=admin&password=admin
```

Figure 2: Authorization request

Reproduction of vulnerability:

```php
/******** AUTHENTICATION *******/
// login / passwords in a PHP array (sha256 for passwords) !
require_once('./passwd.inc.php');


if(!isset($_SESSION['login']) || !$_SESSION['login']) {
    $_SESSION['login'] = "";
    // form posted ?
    if($_POST['login'] && $_POST['password']){
        $data['login'] = $_POST['login'];
        $data['password'] = hash('sha256', $_POST['password']);
    }
    // autologin cookie ?
    else if($_COOKIE['autologin']){
        $data = unserialize($_COOKIE['autologin']);
        $autologin = "autologin";
    }

    // check password !
    if ($data['password'] == $auth[ $data['login'] ] ) {
        $_SESSION['login'] = $data['login'];
```

Figure 3: Vulnerable source code

1. As a result of an unsuccessful authorization attempt, a message is issued from the superadmin user, which suggests that there is a user named superadmin in the system who has high privileges in the system.

2. When authorizing the user, it is checked at the code level whether the login and password parameters have been passed to the POST. If the parameters were passed by the user, it will not be possible to implement the vulnerability.

Line: 18

Code: if($_POST['login'] && $_POST['password']){ ..... }

3. If the login and password parameters have not been passed, then another condition is checked, whether the autologin parameter has been passed to the COOKIE. The body of this condition does not safely deserialize the object passed in the autologin COOKIE parameter.

Line: 23

Code: else if($_COOKIE['autologin']){ ..... }

4. To implement the vulnerability, it is necessary to pass only the autologin COOKIE parameter in the request, which subsequently falls into the deserialization functions. Since the autologin parameter does not pass any filtering, the attacker has the opportunity to pass his own serialized object, thereby changing the values of the data array and its login and password values.

Line: 24

Code: $data = unserialize($_COOKIE['autologin']);

5. After the login and password values of the data array are changed by an attacker through unsafe deserialization, these values fall into the comparison function, which contains a Type Juggling vulnerability, which implies an un-typed comparison between variables. To compare between variables, two equals (==) are used, which is not a safe comparison operator.

Line: 29

Code: if ($data['password'] == $auth[$data['login']]) { ..... }

6. As a result of the transfer of various payloads, it was revealed that if you pass a password equal to one (1), then with an untyped comparison of the unit and the hashed password value, the comparison result will be True and the attacker will be able to log in under the user whose login was transferred without knowing the password of the user himself, in this case superadmin, figure 4.

Payload: a:2:{s:5:"login";s:10:"superadmin";s:8:"password";b:1;}

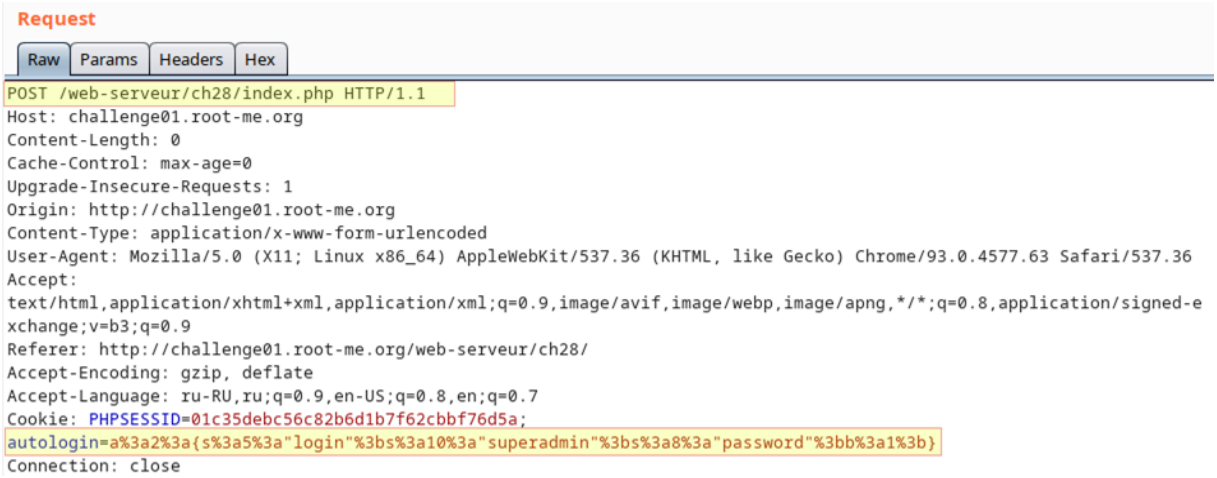

Figure 4: Authorization request with payload

Figure 5: Authorization under superadding

Recommendations for fixing the vulnerability:

1. In this case, the code used for deserialization in the autologin parameter is superfluous and does not participate in the user authorization process, it is recommended to delete this code if it is not used.

2. In the case of using the above code, it is recommended to implement filtering of parameters coming from the user, including filtering from the implementation of serialized objects.

3. Disable Debug mode.

4. To eliminate the Type Juggling vulnerability, it is recommended to use a typed comparison of values, i.e. three equals (===) instead of two (==). When using three equals, in addition to the values of the variables being compared, the data type of the variables being compared will also be checked.

According to this vulnerability report, the auditor is able to identify many inconsistencies in the company's internal processes:

1. The developers of this application made a lot of blunders when developing the authorization panel in the administrative area, which indicates their lack of competence and most likely the absence of regular training courses for developers in the field of information security.

2. There are no automatic source code scanning tools in the organization, this conclusion can be based on information about the popularity of vulnerabilities found during testing, all modern source code scanners are able to identify such vulnerabilities.

3. The absence or incorrect configuration of the Firewall, in this case, the transmitted payload was not noticed and stopped at the stage of exploitation of the vulnerability.

6. Results and discussion

As this example shows, the effectiveness of conducting an audit after conducting penetration testing is a fairly effective measure that is able to identify a number of inconsistencies based on the test results.

Today, many companies are implementing the SecOps methodology, which implies the introduction of automation in the process of ensuring information security. According to the EMA report, this methodology is most often implemented by software developers, banking and financial organizations, wholesale suppliers, manufacturing enterprises, and so on. As a rule, companies that have implemented this methodology increase the following indicators: data quality, awareness of information security issues, improved interaction with users, began to identify the most effective processes within the organization faster. The objectives of the methodology itself include: improving the security level of the organization, uniting teams to improve the management of the organization, raising awareness about information security and how it affects business processes, automating processes through the introduction of reliable tools [22].

The process of implementing the SecOps methodology takes place in several stages: risk audit, risk assessment, verification of cyber hygiene in the organization, integration with business processes. Since after penetration testing, the auditor can identify many non-working or insufficiently well-functioning processes, with the introduction of the SecOps methodology, the previously described processes can be adjusted, and some are automated. For example, based on the results of penetration testing, earlier on the example of the auditor revealed the lack of means in the organization for automatic scanning of the source code, which led to the emergence of unnoticed errors on the part of developers and the possibility of exploiting vulnerabilities on the part of an attacker. The objective of the methodology in this case is to implement automated tools for source code analysis and security scanners to search for vulnerabilities [23], [24]. In addition to the automation already listed, the SecOps methodology implies the introduction of the following tools that will be useful to the organization:

- dashboards – collection and visualization of the collected information, for faster problem finding;
- automation – offline processing of collected data and running algorithms to eliminate errors found (for example, clearing log logs to free up disk space);
- error search – tools for automatic search and error detection;
- sharing – publication of documentation on new implemented technologies;
- visualization – tools for analyzing and viewing data;
- attack modeling – classification of attacks and drawing up a general template that describes the links between attacks and suggests possible solutions.

7. Conclusion

As a result, all the described stages: penetration testing, information security audit and implementation of the methodology of these SecOps should lead to a sharp increase in the information security of the organization. This approach will allow you to identify most of the weaknesses of the organization and subsequently apply the necessary measures to improve the level of information security, establish internal processes and automate them. Having passed all

the necessary stages, we get an organization that currently has not only a high level of protection, but also all the necessary tools for early detection of threats and automated, working internal processes. In further research, it is planned to identify the main parameters in penetration testing, which will help in the future to create a model of a neural network for conducting an information security audit in various organizations. In a white-box penetration testing scenario, a qualitative approach can bring significant gains in identifying vulnerabilities. Qualitative methods can provide valuable insights into the internal workings of a system, including its design, architecture, and configuration. This information can be used to identify potential security weaknesses that might not be obvious from a purely quantitative analysis. Additionally, qualitative approaches can be used to assess the potential impact of a vulnerability on the system and its users. For example, in-depth interviews with system administrators and users can provide valuable information on how the system is used, the types of data stored on it, and the ways in which users interact with it. This information can be used to understand the risks posed by a vulnerability and prioritize which vulnerabilities need to be addressed first.

The scientific novelty of the white box method in web application penetration testing lies in its ability to provide a more comprehensive and targeted approach to identifying vulnerabilities within an application. By utilizing knowledge of the internal workings of the application, the white box method allows for a deeper understanding of the system and its potential vulnerabilities.

This approach stands out from other methods of web application penetration testing due to its effectiveness in identifying vulnerabilities early in the development process. By detecting and remediating vulnerabilities early on, the cost and time associated with addressing vulnerabilities after deployment can be reduced. The white box method also enables developers to gain a better understanding of the security implications of their code, leading to more secure software and better overall security practices.

Overall, the scientific novelty of the white box method lies in its ability to provide a more comprehensive and targeted approach to web application penetration testing, leading to improved security and reduced costs and time associated with vulnerability detection and remediation.

Finally, qualitative approaches can also help to identify the root cause of a vulnerability, which is critical in developing a comprehensive and effective remediation strategy. By gaining a deep understanding of the system and its design, testers can determine the underlying cause of a vulnerability and develop a targeted solution that addresses the root issue, rather than simply patching the symptom.

In conclusion, incorporating qualitative methods into a white-box penetration testing methodology can bring significant benefits in terms of identifying vulnerabilities, assessing their potential impact, and developing effective remediation strategies.

8. References

[1] S. Gnatyuk , Critical Aviation Information Systems Cybersecurity, NATO Science for Peace and Security IOS Press Ebooks, 47 3 (2016) 308-316.

[2] A. Aibekova, V.Selvarajah Offensive Security: Study on Penetration Testing Attacks, Methods, and their Types (2022) doi:10.1109/ICDCECE53908.2022.9792772.

[3] M. Slunjski, D. Sumina, S Groš; Igor Erceg Off-the-Shelf Solutions as Potential Cyber Threats to Industrial Environments and Simple-To-Implement Protection Methodology. (2022) doi: 10.1109/ACCESS.2022.3217797.

[4] N. A. Almubairik; G. Wills Automated penetration testing based on a threat model. (2017) doi: 10.1109/ICITST.2016.7856742

[5] S. J. Stouffer, S. R. Bellovin, M. W. Hoehn, and J. A. Hunker, Guide to penetration testing: Testing network security, NIST Special Publication, 800-115 (2013).

[6] E. M. Okonigene, A. O. Ilori, and O. B. Longe, "Penetration testing: A critical review," Journal of Emerging Trends in Computing and Information Sciences, 7 12 (2016) 621-629.

[7] C. V. Nguyen and N. Nguyen, A survey of penetration testing techniques, Proceedings of the International Conference on Advanced Computing and Applications, 2016.

[8] J. I. Adan and M. A. Awad, Penetration testing methodology for mobile application, Proceedings of the International Conference on Computer, Control, Informatics and Its Applications, 2017.
[9] A. Adhikari and M. Shah, A comprehensive study on penetration testing: Methods, tools, and techniques, Proceedings of the International Conference on Computer Communication and Informatics, 2017.
[10] C. T. C. Machado, D. A. O. Cunha, and M. S. Ribeiro, A systematic mapping study on penetration testing methods and tools," Journal of Information Systems Engineering & Management, 3 4 (2018) 1-18.
[11] S. S. Al-Maashri and I. A. Al-Salti, A review of penetration testing methodologies, International Journal of Advanced Computer Science and Applications, 11 2 (2020) 101-108, 2020.
[12] M. R. Nikam and D. D. Doye, A comparative study of penetration testing methodologies, Proceedings of the International Conference on Communication and Signal Processing, 2018.
[13] M. A. Khan and M. Y. Javed, Penetration testing methodologies and tools: A review, Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies, 2019.
[14] S. S. Al-Maashri and I. A. Al-Salti, "A comparative study of penetration testing methodologies," International Journal of Advanced Computer Science and Applications, 11 4 (2020) 68-73, 2020.
[15] M. A. Al-Saleh, Penetration testing: Methods, tools, and techniques, Proceedings of the International Conference on Computing and Communication Technologies, 2016.
[16] N. R. Jaiswal and R. K. Dubey, A comparative study of penetration testing methodologies, Proceedings of the International Conference on Intelligent Computing and Control Systems, 2018.
[17] S. Raval and N. Bhavsar, Comparative analysis of penetration testing methodologies, International Journal of Computer Science and Information Security, 17 8 (2019) 7-14.
[18] A. M. Hasan and F. N. Al-Khasawneh, A comprehensive study of penetration testing methodologies, International Journal of Advanced Computer Science and Applications, 9 5 (2018) 20-27.
[19] S. A. Khan and A. I. Kawsar, Penetration testing methodologies: A comprehensive review, Journal of Information Technology Research, 10 3 (2019) 486-503.
[20] T. C. V. C. Souza and F. A. T. de Carvalho, A systematic review on penetration testing methodologies for web applications, Journal of Information Systems Engineering & Management, 4 2 (2019) 35-47.
[21] M. R. Nikam and D. D. Doye, A review on penetration testing methodologies for network security, Proceedings of the International Conference on Recent Trends in Computer Science and Electronics Engineering, 2018.
[22] M. Alqhtani, A. Alharthi, and M. Alshamrani, A review on penetration testing techniques and methods for network security, Proceedings of the International Conference on Future Networks and Distributed Systems, 2019.
[23] A. Alqurashi and M. Alharthi, A comparative study of penetration testing methodologies for network security, in Proceedings of the International Conference on Advanced Information and Communications Technology, 2019.
[24] Y. Duan, Z. Han, and Y. Shen, "Research on penetration testing methods for security evaluation of intelligent buildings, Journal of Ambient Intelligence and Humanized Computing, 11 5 (2020) 2125-2136.