UDC 004.41
IRSTI 81.93.29
https://doi.org/10.55452/1998-6688-2023-20-2-125-136

Samburskaya S.A.
International information technology university 050040, Almaty, Kazakhstan
E-mail: sofiya.samburskaya@gmail.com

DEVELOPMENT OF A SECURE LOGGING AND MANAGEMENT SYSTEM FOR PENETRATION TESTING

Abstract. The sphere of information security in Kazakhstan affects an increasing number of industries every year, and penetration testing is also gaining popularity, as it is one of the key methods for assessing the security and risks of a company. This article is devoted to the research and development of a web application to provide full control over the penetration testing process: monitoring the implementation of tasks and projects, reporting on all processes, dividing tasks between employees. The management system automatically selects recommendations for eliminating vulnerabilities and generates reports on penetration testing. As a classification algorithm, a decision tree is used. Differentiation of users by access levels, structured data storage, automatic recording of test results, generation of reports and selection of recommendations for eliminating vulnerabilities make the web application more perfect and convenient compared to similar systems. The importance of this study lies in the simplification of the implementation of penetration testing and the development of this service in Kazakhstan, which will improve the level of information security in enterprises of all industries.

Key words: web application, management system, html, css, postgresql, penetration test, django.

Самбурская С.А.
Халықаралық ақпараттық технологиялар университеті 050040, Алматы қ., Қазақстан
E-mail: sofiya.samburskaya@gmail.com

ҚАУІПСІЗ КАРОТАЖ ЖӘНЕ БАСҚАРУ ЖҮЙЕСІН ДАМЫТУ ЕНУ СЫНАҒЫ ҮШІН

Аңдатпа. Қазақстандағы ақпараттық қауіпсіздік саласы жыл сайын өнеркәсіптердің көбеюіне әсер етеді және енуді тестілеу де танымал болуда, өйткені ол компанияның қауіпсіздігі мен тәуекелдерін бағалаудың негізгі әдістерінің бірі болып табылады.
Бұл мақала ену тестілеу процесін толық бақылауды қамтамасыз ету үшін веб-қосымшаны зерттеуге және әзірлеуге арналған: тапсырмалар мен жобалардың орындалуын бақылау, барлық процестер туралы есеп беру, қызметкерлер арасында тапсырмаларды бөлу. Басқару жүйесі осалдықтарды жою бойынша ұсыныстарды автоматты түрде таңдайды және енуді тексеру туралы есептерді жасайды. Жіктеу алгоритмі ретінде шешім ағашы қолданылады. Қолдану деңгейлері бойынша пайдаланушыларды саралау, құрылымдық деректерді сақтау, сынақ нәтижелерін автоматты түрде жазу, есептерді құру және осалдықтарды жою бойынша ұсыныстарды таңдау ұқсас жүйелермен салыстырғанда веб-қосымшаны мінсіз және ыңғайлы етеді.
Бұл зерттеудің маңыздылығы еніп кетуді тестілеуді енгізуді жеңілдетуде және Қазақстанда осы қызметті дамытуда, бұл барлық сала кәсіпорындарында ақпараттық қауіпсіздік деңгейін арттыруға мүмкіндік береді.

Тірек сөздер: web-қолданбалар, басқару жүйесі, html, css, postgresql, ену сынағы, django.

Самбурская С.А.
Международный университет информационных технологий, 050040, г. Алматы, Казахстан
E-mail: sofiya.samburskaya@gmail.com

РАЗРАБОТКА БЕЗОПАСНОЙ СИСТЕМЫ ЛОГИРОВАНИЯ И УПРАВЛЕНИЯ ДЛЯ ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ

Аннотация. Сфера информационной безопасности в Казахстане с каждым годом затрагивает все большее количество отраслей, и тестирование на проникновение также набирает популярность, так как является одним из ключевых методов оценки безопасности и рисков компании. Данная статья посвящена исследованию и разработке веб-приложения для обеспечения полного контроля над процессом тестирования на проникновение: мониторинг выполнения задач и проектов, ведение отчетов по всем процессам, разделение задач между сотрудниками. Система менеджмента автоматически осуществляет подбор рекомендаций по устранению уязвимостей и генерирует отчеты о проведенном тестировании на проникновение. Как алгоритм классификации используется древо решений. Разграничение пользователей по уровням доступа, структурированное хранение данных, автоматическая фиксация результатов тестирования, генерация отчетов и подбор рекомендаций по устранению уязвимостей делают веб-приложение более совершенным и удобным на фоне аналогичных систем.

Важность этого исследования заключается в упрощении осуществления тестирования на проникновение и развития этой услуги в Казахстане, что позволит усилить уровень информационной безопасности на предприятиях всех отраслей.

Ключевые слова: веб-приложение, система управления, html, css, postgresql, тест на проникновение, django.

Introduction

The first mention of the creation of a team for penetration testing dates back to the 1970s. It can be argued this area is still young and developing.

Penetration testing is now offered as an IT service, and any IT service can be considered a project. In order to facilitate and systematize project management, it is proposed to create a web application, taking into account the peculiarities of penetration testing.

The results of the study showed that in most companies it is possible to access resources on the local network. This proves the development potential of the Offensive Security area.

Developing a web-based management system for penetration testing with logging abilities will help ensure complete control over the penetration testing process: monitoring tasks execution, keeping reports of all processes, division of tasks between employees.

The goal of this article is to develop a secure logging and management system for penetration testing and validate its effectiveness in a simulated penetration test.

To achieve the set goal, it is necessary to focus on the following research tasks:

- develop a web application for the management of penetration testing;
- develop module for web application for logging functionalities;
- create module for secure connection to database.

Main provisions

Penetration test is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

The penetration testing service is divided into several types:

- network service tests;
- web application tests;
- client-side tests;
- wireless network tests;
- social engineering tests.

Network service test is the widest direction of penetration testing. It aims to discover vulnerabilities and attack vectors inside the client's company infrastructure. It can be subdivided into external and internal, and methods for these subdivisions differ as well. The network of the client can be tested from outside, simulating outer threat actor, and from inside, simulating insider threat.

Web Application Test is a project of discovering vulnerabilities in the web application itself, their plug-ins, applets, Content Management System (CMS), web-server, database and etc. Sometimes, it also includes an analysis of the source code of web application. In the context of web application security, penetration testing is commonly used to improve the web application firewall (WAF).

Client-side penetration testing is the act of exploiting vulnerabilities in client applications such as email clients, web browsers, office applications, and others. It is also called internal testing. The purpose of these tests is to identify security threats that arise locally.

A wireless penetration test emulates an attacker trying to gain access to the internal network through the wireless network. It is conducted to assess the adequacy of a variety of security measures designed to protect against unauthorized access to wireless services.

Social Engineering Penetration Testing is an attempt to gain some level of access using social engineering on company employees to determine the level of awareness of the organization's employees.

A client can reach out for several types of penetration testing or only one depends on the needs and level of the IT infrastructure of their company.

At the end of any penetration testing project, report is submitted and presented to the client.

A study in the field of financial losses of companies as a result of information attacks showed that, on average, one company loses up to $ 1 million due to cyber-attacks. Such a blow to the company's budget leads to consequences in the form of a drop in the value of the company's shares by an average of 7.27%.

In addition, a comparative analysis of the cost of conducting penetration testing was carried out. During the analysis, it was found that the average cost of a pentest service is from 10 to 30 thousand dollars.

Based on these data, it is concluded that pentesting is a very profitable investment for companies. Such an investment will allow not only to detect possible threats to the company's information resources, but also to avoid the likely large financial losses that often accompany cyber-attacks.

Materials and methods

Penetration testing is now offered as an IT service, and any IT service can be considered a project. To facilitate and systematize project management, a web application was proposed to be created, considering the peculiarities of penetration testing.

The main features of the developed system are presented below ai the Figure 1.

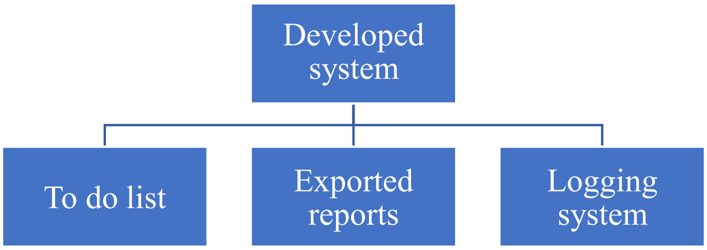

Figure 1 – Main features of developed system

The developed system will assist in the management of penetration testing projects by presenting an opportunity for the project manager or team leader to create new projects and subdivide them into tasks and then assign these tasks to pentesters, who will be also users of this web-application.

The process of generating reports will be automated and based on analytics and pieces of evidence, which will be added by pentesters during testing will be parsed from tasks.

For easier capturing the shreds of evidence logging script is presented, which will be launched on pentesters workstation and will collect data in the console.

The following software will be used to develop web application and logging system: Python, Django Framework, PostgreSQL Database. Python will be used to build the registration system.

Django is a high-level Python web framework that promotes fast development and clean, pragmatic design. Django makes it easy to build better web applications quickly and with less code.

Django provides a bridge between the data model and the database engine, and supports a wide range of database systems, including PostgreSQL.

PostgreSQL is not just a relational, but an object-relational DBMS. This gives it some advantages over other open-source SQL databases such as MySQL, MariaDB and Firebird.

Django and PostgreSQL will be used to build a web application. This software was chosen because Django, as a framework designed specifically for web development, is secure and has all the features and libraries needed to fulfill the goals of this project.

In addition, Django is SQL injection-proof and the combination of Django and PostgreSQL in web development is considered the best tandem.

Also, Django has feature that will help expand functionality of the developed web application in the future. Django's work structure makes it easy to embed new feature branches into the finished application without any complications.

Flowchart of web application operation is presented on the Figure 2.

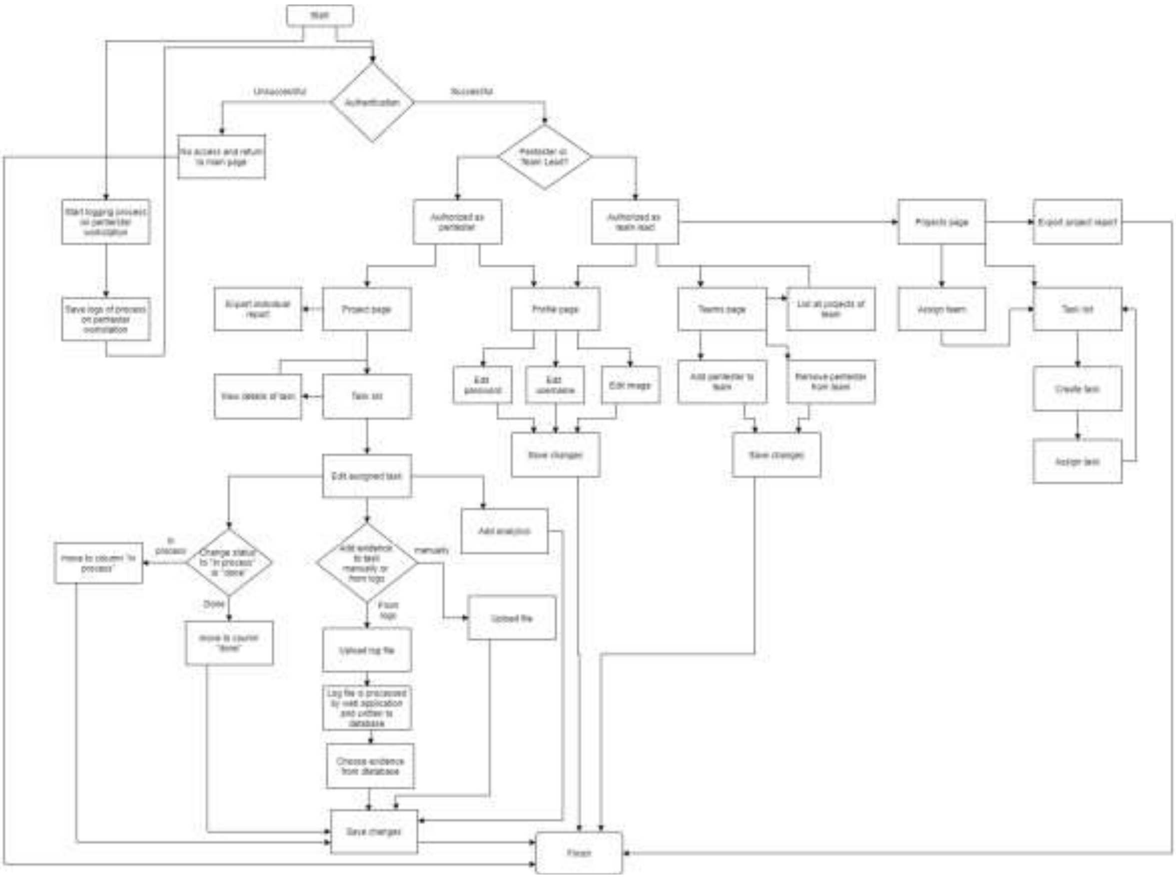

Figure 2 – Flowchart

The flowchart can be divided into several main functions of web-application:

- authentication;
- authorization;
- viewing and editing profile page;
- viewing, creating, and editing teams;
- viewing, creating, and editing projects;
- viewing, creating, and editing tasks of the project;
- work of logging script;
- process of exporting of individual and/or project report.

The flowchart also reflects the differences in the access of the pentester and the team lead.

Scenario of work of the system depends on the role of the user. User can be assigned as team lead of the project or as pentester. User can perform task, which needs to be performed in series, or individual tasks as well in the system. Use case diagram is presented om the Figure 3.

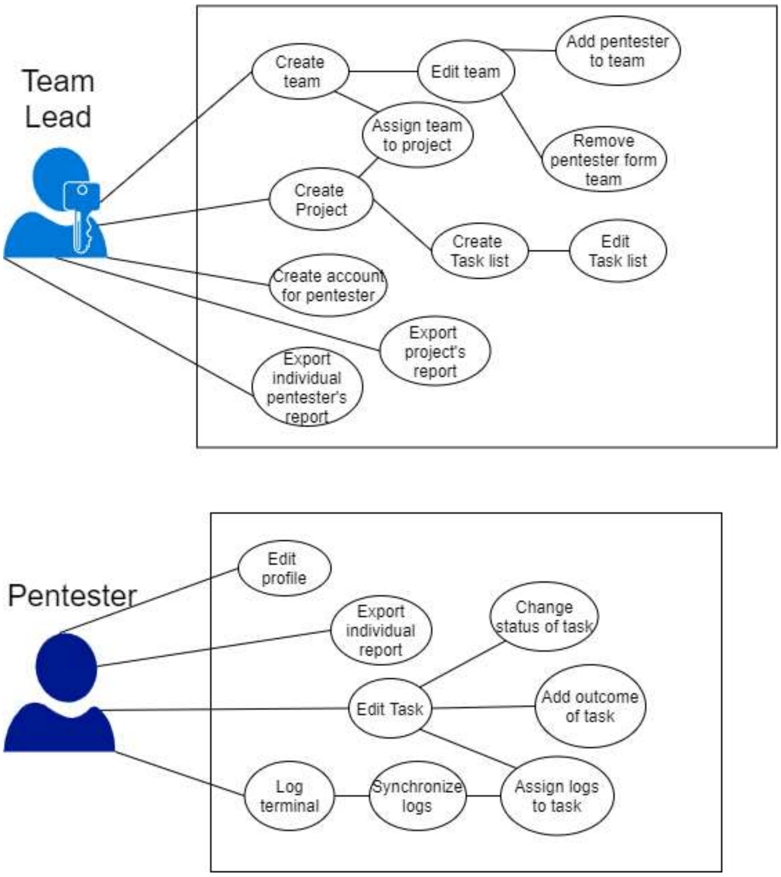

Figure 3 – Use Case diagram

This diagram represents a set of activities possible in the system based on the role of the user.

In the case of team lead, the user can perform management functions. The team lead can form a team of penetration testers for the project and, if necessary, can edit the existing team by removing or adding pentester.

Next user can start the process of creating a new project, which includes assigning a team of pentesters and creating a task list.

As the system will be presented as a platform and will be distributed as a subscription, the team lead will register their new pentester inside the system.

At any time, user can export a report of work of individual pentester for tracking the results.

At the end of the project, the team lead can export the whole project's report in form of a pdf file.

In the case of pentester, the user can work on an assigned task and as the result, change the status of the task to “done” or “in-process”, add analytics of the results of performed tasks and add evidence of existing vulnerabilities.

Besides, the user can launch script, which will collect logs from pentester's terminal.

Also, pentester can export a report of their individual work.

User can edit their personal data in the profile.

This UML diagram shows the step-by-step interaction between a user, a web application, and a database. The diagram shows the sequence of all activities over time.

Processing sequence is illustrated below on the Figure 4.

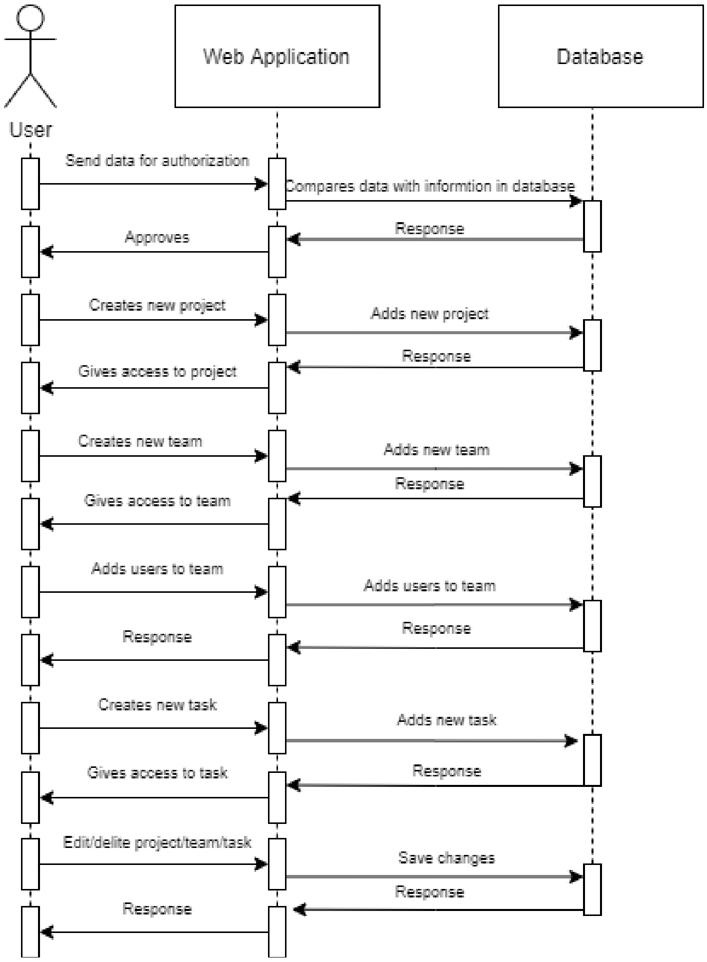

Figure 4 – UML diagram

First, the user connects to the web application and enters his credentials. The application sends a query to the database to compare the information entered by the user with the information stored in the database. The database gives a response to the application, the application allows the user to access the application's functionality. After authorization, the user can create projects, teams, and add users to teams. To carry out these actions, the user enters the necessary information, the application sends a request to the database to save the information. The information is stored in a database.

The diagram of database structure is presented on the Figure 5.

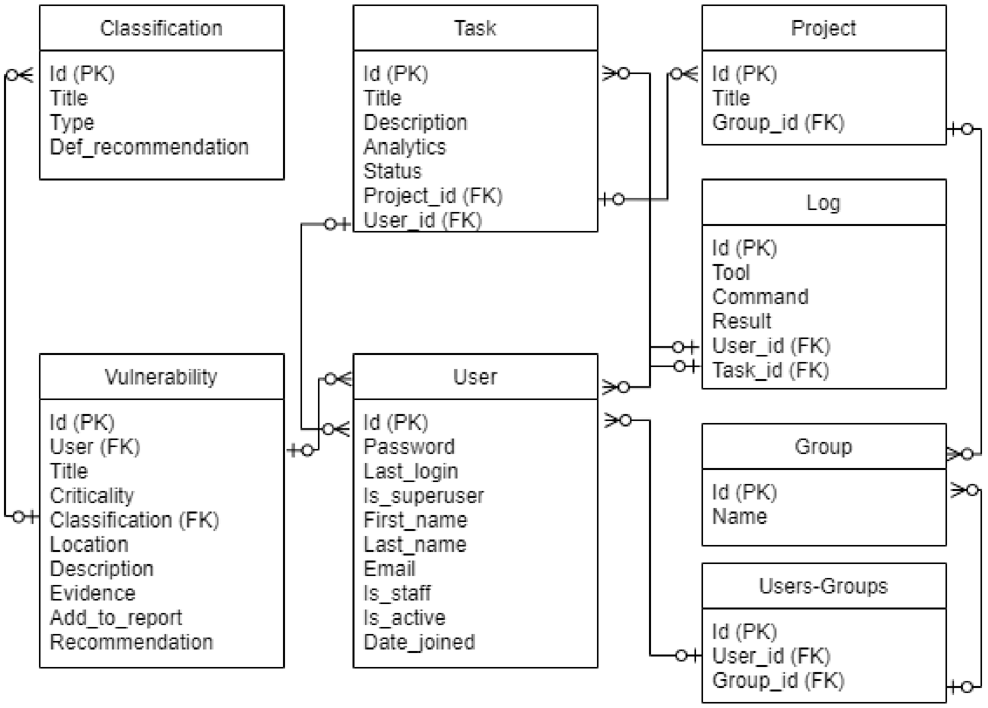

Figure 5 – Database structure

This diagram shows the structure of the web application database. The database includes the following tables: user, group, user-groups, project, task, classification, vulnerability, log. All tables are linked by a one-to-many relationship. The relationship between tables is carried out utilizing a foreign key.

The User table stores data about all users of the web application, from this table the user id is transferred to other tables, where the user is specified as a foreign key.

The Group table includes information about all groups exist in the web application.

The User-Groups table is the link between tables User and Group. All data from this tables (user id, group id) is passed as a foreign key.

The Project table stores information about all projects that are in the web application. A group is attached to the project, which is assigned to these projects. This is done by passing the group id from the "Group" table as a foreign key.

The Task table contains the following information for the task: title, project to which the task belongs, task status, description, analytics, and the user who is assigned responsibility for this task. All data from other tables (user id, project id) is passed as a foreign key.

The Classification table stores data about vulnerabilities' classification: title, type, recommendation.

The Vulnerability table stores data about all founded vulnerabilities. All data from other tables (user id, classification id) is passed as a foreign key.

The Log table include following columns: tool. command, result, user id, task id. All data from other tables (user id, task id) is passed as a foreign key.

The system architecture is presented below on the Figure 6.

This diagram depicts the architecture of the system and shows the protocols and ports through which communication is carried out between the end user's computer and the server that stores the web application and database.

The end user's computer communicates with the server that hosts the web application and database over TCP over port 443.

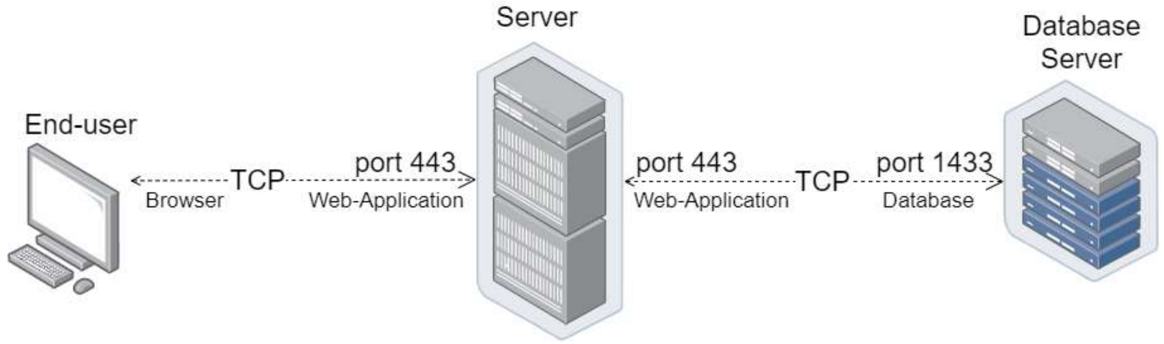

Figure 6 – System architecture

Through this protocol, the end user interacts with the server throughout the functionality of the web application. The server sends a response to requests in the same way. This interaction makes the user work correctly with the web application. The login script runs on the end user's computer. After completing all tasks, the collected information is saved and sent to the database via TCP on port 1433 and saved in the database.

Results and discussion

Before proceeding with the development, it was necessary to select the optimal tools for its implementation. When choosing the tools, the relevance, work skills, compatibility of tools with each other were considered, since the project includes integration between a database, a script for collecting logs, backend and frontend parts of the program.

Below is a list of all the programs that were used in the development of project:

- PostgreSQL is SQL based object-relational DBMS;
- pgAdmin is an administration and development platform for PostgreSQL and related database management systems. The platform supports all PostgreSQL features. The convenient and simple interface of the platform made it possible to work on the project in terms of the database quickly and efficiently;
- Visual Studio Code a source code editor that is supported on Windows, Linux and macOS operating systems. It is a code editor for cross-platform web and cloud application development. Includes a debugger, syntax highlighting, and other benefits that were useful during project development;
- Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing, used by professionals worldwide due to a large collection of tools. Kali Linux was used as an example of the working environment of a penetration tester.

Presented web-application is created and deployed using several files:

- settings.py is created by Django framework and consists of main settings of web-application, including credentials and connection details to the database, declaration of global variables and IP address of web-application;
- urls.py is presented as a mapping between URL path expressions to views functions;
- models.py is a collection of Python objects, which are used for easier access and managing during work of web-application. Models set the structure of stored data, including the field types;
- views.py is used for creating Python functions for processing a web request and returning a web response;
- forms.py is used to create HTML forms and declare their work and appearance;
- template is a collection of HTML pages that will be returned as a response to the web-application.

HTML and CSS were used to develop the frontend. Since the end user of the web application will interact with him precisely through the frontend, it was necessary to make it convenient and understandable for users.

Also, during the development of the frontend part, it was considered that the users will work with the web application on various devices, so the front-end of the project is made adaptive for desktop version, mobile phones, and tablets. Due to adaptivity of front-end part it will be comfortable for user to work with web application from any device during assignment.

Since the pentest management system is positioned in the future as a paid platform, the user cannot register in the system on his own. For initial use, the end user must already have an account for authorization.

The Django model is a built-in function that Django uses to create tables, their fields, and various constraints, simplify tasks, and organize tables in a model. Each model is mapped to one database table.

The first model created for the web-application was the Project model. This model will be used as a structure for the creation of future penetration testing projects. It can be created by filling with information fields as:

- Title is displayed the name of the penetration testing project, which will be used in menus and will be written in the generated report;
- Team is presented as a foreign key to the table of a team of penetration testers. This field will be used for future authorization checks.

The next is a Task model, which is used for the creation of duties' subdivision in penetration test project. The structure of this model is presented as such fields as:

- Title displays the name of the task;
- Description is a short declaration of needed work and expected results, written by team lead or project manager;
- User is a foreign key to the users' table and is used to specify who is responsible pentester for this task;
- Analytics is a short description of the results of the task for future use during the project;
- Status defines if the task is on a to-do list, in-process list, done list.

Based on the object created with vulnerability model report for the project is generated. The creation of this object is carried out by filling such fields:

- Project is a foreign key for determining to which project belongs this vulnerability;
- User is a foreign key for determining the user who added this vulnerability to the database;
- Title displays the name of vulnerability;
- Criticality determine to which levels this vulnerability belongs: low, medium, or high;
- Classification is text stings for adding a note about the classification of vulnerability based on standards, for example, OWASP Top-10;
- Location determines the place where this vulnerability exists;
- Description is a declaration of what this vulnerability represents, how it was found, and possible threats of this vulnerability;
- Evidence is a file that can prove the existence of this vulnerability;
- Add to report is a checkbox to determine whether vulnerability will be included in the report;
- Recommendation is a declaration on how to eliminate this vulnerability.

The next is a Classification model, which is used for simplified vulnerability report. The creation of this object is carried out by filling such fields:

- Title displays the name of classification;
- Type - selected from the list of existing classifications of vulnerabilities: web, network. mobile, physical or configuration;
- Recommendation is text stings for adding a recommendation for eliminating vulnerabilities.

The next is a Log model, which is used for storing logs from the terminals of pentesters, which were recorded during the work on the project. The creation of this object is carried out by filling such fields:

- Tool displays the name of used tool;
- Command displays the full command that the pentester introduced during the work on the project;
- Result displays full output of used command;
- User is a foreign key for determining the user who added this log to the database;
- Task is a foreign key for determining the task in the performance of which was recorded by the log.

The first page of the web application that the user comes to after registration is my profile. The functionality that is available to the user in the personal account differs depending on the access level.

The primary information that the user sees in a profile - a username, email address, access level, and projects assigned to the user's team.

Functionality that is available to the user after authorization:

- team leader - creating projects, creating teams, creating accounts for pentesters, adding pentesters to a team, editing tasks, adding vulnerabilities, generating a report;
- pentester - editing tasks, adding vulnerabilities, generating a report.

Scenario of work of the system depends on the role of the user. User can be assigned as team lead of the project or as pentester. The user of the web application with the status of team leader has access to the functionality of creating accounts for pentesters. This functionality provides the team leader's control over employees, the ability to grant or deny a certain user access to the web application.

Functionality of creating a team and adding pentesters to it provides control over the work on the project, distribution of workload and responsibilities between employees. After successfully creating a team, it is necessary to add users to it. In addition, the distribution of pentesters into teams makes it possible to control employee access to projects and prevent employees who do not take part in the implementation of a specific project from accessing projects.

The user of the web application with the status of team leader has access to the project creation functionality. When adding a new project, its name is indicated and a team of pentesters is attached, which will be responsible for the implementation of the project.

The functionality of creating the task is implemented using the Djando form and using the Django view. The form on the web-page is generated with fields of add_task form. Then data is sent using POST and processed with add_task view. In case when all fields are valid, a new task of the project is added to the database.

The functionality of adding a task is implemented as a form on the web page.

During the project, an authorized user can see a list of current tasks divided into three categories based on their status: to do, in-process, and done. To see details of each task user should click on the card of the needed task.

After accessing a page of the task, the authorized user can see details about this vulnerability and there is an update button if this task belongs to this user or this user is the project manager.

If the user is authorized to edit this task, the web application responds with a form.

If the user is a team-lead or project manager of the project, then they can add new tasks and update existing ones during the penetration testing project.

If the user is authorized to access the project by belonging to the owner-team, then they can view the list and view details of existing tasks of the project, also they can update tasks assigned to them.

If the user is authorized to access the project by belonging to the owner-team, then they can add, view the list, and view details of vulnerabilities discovered during the penetration testing project.

The functionality of creating of record about vulnerability is implemented using Django form and using the Django view. The form on the web-page is generated with fields of add_vulnerability form. Then data is sent using POST and processed with add_vulnerability view. In case when all fields are valid, a new record about vulnerability is added to the database.

Functionality is presented in simple form on the web-page.

After filling in the basic information about the vulnerability, the user will be redirected to the form for recommendations about the elimination of this vulnerability. If a default recommendation was written during adding a classification, it will automatically fill in the generated form field for the current vulnerability with the ability to edit. This will facilitate the process of writing the report.

During the project, an authorized user can see a list of currently found vulnerabilities divided into three categories based on their criticality: low, medium, and high. To see details of each vulnerability user should click on the card of needed vulnerability.

After accessing a page of vulnerability, the authorized user can see details about this vulnerability and download a file, which is served as evidence.

After accessing the web-page dedicated to the generation of the report and choosing a needed project, the user clicks on the button “Generate”. This will lead to the generation of the report with random names and availability to download this report to the user’s workstation.

The report is presented as a HTML file.

Generation of reports is implemented using download report view, which firstly writes all details of vulnerabilities to the file and then responds to the user with file’s content.

If the user is a project manager or team-lead, they can access the functionality of the report generation for their penetration testing projects.

To simplify the presentation and exchange of the results of task execution, the logging functionality is presented, which is a Python script. Before starting the task, the pentester launches a script that creates a folder for the current project and launches the Linux system command "script" to record input and output terminal data. After finishing work on the task, the pentester enters the "exit" command, ending the script execution. The data file will be stored in the current project folder.

Next, the pentester can attach a file with logs to the completed task through the form.

After adding the file, the system processes the logs using the view. The system first checks if the task belongs to the current user as part of the access controls. If successful, the web application checks if the file is a text file that is expected as a result of the script. After all checks, the web application saves the file with a random name and processes it for further parsing.

Parsing is carried out by terminal key symbols and words, and the parsing results are saved to an array. Then the contents of the array are saved as instances of the Log model.

Conclusion

As a result of study, a web application for secure logging and a management system for penetration testing was developed. This system helps ensure complete control of the process of penetration testing: monitoring tasks execution, keeping reports of all processes, division of tasks between employees.

The functionality of the web application was specially designed to be suitable for the needs of penetration testers and project managers.

During the development were faced and executed work with international classification of vulnerabilities, encryption of personal data, work with the database, and development the web application and logging script.

Currently, requests for penetration testing are coming from companies in the following industries: finance, information technology, government, energy and fuel, medicine, entertainment, telecommunications, and industry. The number of requests from companies in the financial sector for penetration testing service is the highest.

Web application and logging script were implemented in the Python programming language. The virtual machine with Kali Linux running as a server for system was used, which is used by professionals to perform penetration testing, so it is close to real project conditions. The front-end was made using CSS and HTML. The Back-end part was written in Django. Also, PostgreSQL was used as a database.

Effectivity of the usage of the web application was proved during the simulation of a penetration testing project using vulnerable virtual machines. During this simulation work of the team of three professionals was organized, execution of task was logged, and the report was generated based on entered into web application entries about vulnerabilities.

There is still a space for improvement for the current project, such as introduction of different language versions or generation of reports in different formats, but work on the development will be proceeded, which will lead to updating the current version or the creation of different versions.

References

1 Canadian Centre for Cyber Security, Cyber Threat and Cyber Threat Actors [online]. ISBN 978-0-660-45950-9. https://www.cyber.gc.ca/sites/default/files/ncta-2022-intro-e.pdf (2020).

2 Verizon, 2019 Data Breach Investigations Report [online]. https://www.nist.gov/system/files/documents/2019/10/16/1-2-dbir-widup.pdf (2019).

3 Bischoff P. (2020) How data breaches affect stock market share prices.

4 Aileen G. Bacudio, Xiaohong Yuan, Bei-Tseng Bill Chu, Monique Jones (2011) An Overview Of Penetration Testing, International Journal of Network Security & Its Applications, vol.3, no.6.

5 Positive Technologies, External pentests results – 2020, Penetration testing of corporate information systems (2020).

6 Hessa Mohammed Zaher Al Shebli, Babak D. Beheshti, A study on penetration testing process and tools, IEEE Long Island Systems, Applications and Technology Conference. https://doi.org/10.1109/LISAT.2018.8378035.

7 Chiem Trieu Phong, Wei Qi Yan, An Overview of Penetration Testing, International Journal of Digital Crime and Forensics, 25. https://doi.org/10.4018/ijdcf.2014100104.

Information on the author

Samburskaya Sofiya Alexandrovna

Master student of the Computer Engineering Department, Tutor of the Cybersecurity Department of the International information technology university, Manas st., 34, 050040, Almaty, Kazakhstan

ORCID ID: 0000-0003-1257-6368

E-mail: sofiya.samburskaya@gmail.com.

Автор туралы мәлімет

Самбурская София Александровна

«Компьютерлік инженерия» кафедрасының магистранты, Халықаралық ақпараттық технологиялар университетінің киберқауіпсіздік кафедрасының тәлімгері, Манас, 34, 050040, Алматы қ., Қазақстан

ORCID ID: 0000-0003-1257-6368

E-mail: sofiya.samburskaya@gmail.com.

Информация об авторе

Самбурская София Александровна

Магистрант кафедры «Компьютерная инженерия», тьютор кафедры «Кибербезопасность» Международного университета информационных технологий, ул. Манаса, 34, 050040, Алматы, Казахстан

ORCID ID: 0000-0003-1257-6368

E-mail: sofiya.samburskaya@gmail.com.