Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 3 (2023) 705 - 715 ISSN Media Electronic: 2580-0760

Remote Penetration Testing with Telegram Bot

Naufal Hafiz[1], Obrina Candra Briliyant[2], Dimas Febriyan Priambodo[3*], Muhammad Hasbi[4] , Sri Siswanti[5]
[1,2,3]Cyber Security Engineering, National Cyber and Crypto Polytechnic
[4,5]Informatic, STMIK Sinar Nusantar
[1]naufal.hafiz@bssn.go.id, [2]obrina.candra@poltekssn.ac.id, [3]dimas.febriyan@poltekssn.ac.id, [4]mhasbi@sinus.ac.id, [5]syswanty@gmail.com

Abstract

The widespread of websites and web applications makes them the main target of cyber attacks. One way to increase security is to perform a penetration test. This test is carried out using the attacker's point of view to find out vulnerabilities on a website or web application and then exploit these vulnerabilities. The results of the penetration test can be used as recommendations to close the gaps that have been known through testing. Because penetration testing requires special resources such as tools and operating systems, a solution is needed to make penetration testing possible with low resources. Telegram bots that are open source offer a solution to overcome these problems. Using the SDLC waterfall approach, this bot was built to provide penetration testing services by connecting the Kali Linux server as a tools provider and the Telegram bot as an interface to users. As a result, users can access penetration testing tools anywhere and anytime via the Telegram bot. To ensure that the bot can run well, testing is carried out through black box testing and load testing. Telegram bot is a solution for integrated compact automatic mobile penetration tester with low resources. Based on load testing, the maximum limit of users who can access Telegram bots simultaneously is 35 users with the highest load average of 5.4. Based on the results of the User Acceptance Test, the Telegram bot has an acceptance rate score of 88,457 % and a questionnaire score of 774 which is an agreed area.

Keywords: *mobile penetration tester; penetration testing; telegram bot; web vulnerabilities.*

1. Introduction

The trend of using cyberspace is increasingly widespread and cannot be separated from everyday life. Starting from sharing information, work, learning, business, and other services. This triggers excessive use of websites and web applications in all fields, making them one of the targets of cyber attacks [1]. Based on the IBM report in 2020, there was an increase in cybersecurity incidents such as data theft which increased by 160%, illegal server access by 233% and scan-and-exploit-based attacks which accounted for 35% of attack types in 2020 [2]. Regardless of the type of attack, a target in the form of a website, network, or information system in an organization will always be exploited with various attacks if it still has vulnerabilities [3]. One way to find out the vulnerability of an existing system is to use a penetration test.

Penetration testing is a security evaluation from the attacker's point of view to test existing vulnerabilities by exploiting [4]. The goal is to conduct a security assessment based on the type of attack with various penetration tools [5]. Due to the large number of tools needed, the test requires large resources such as tools, operating systems, and devices which are not owned by everyone [6]. Whereas penetration testing is very important to determine the security of an asset against attacks. Therefore, a solution is needed in conducting penetration tests with low resources.

Telegram is one of the most popular multi-platform instant messaging applications in the world [7]. Since its launch until 2021, Telegram has had 500 million monthly active users and is among the 10 most downloaded apps of 2020 [8]. This is supported by the many features provided by Telegram such as private chat, channels, and bots. The Telegram bot feature is quite useful because it is open source so that it can be made according to user needs as an automated service [9]. The bot will respond to messages from users or clients in the form of commands to be forwarded to the server then the server will provide the requested service [10]. Telegram bots can also be a means to perform penetration tests because bots can be connected to the tools provider server. For example Remote-shell bot which provides Linux terminal facilities to perform simple commands on native Linux via Telegram chat

Accepted: xx-05-2023 | Received in revised: xx-05-2023 | Published: xx-06-2023

[11]. There is also Telegram-bot-for-pentest which provides enumeration tools for port and DNS scanning [12].

Remote-shell bot and Telegram-bot-for-pentest programs allow work to be done remotely. This remote trend allows users to do their work remotely through a platform that can access resources [13]. According to a report from Owl Labs, 16% of global jobs are done remotely, and 70% of workers are working remotely during the pandemic [14]. By implementing remote, users can work flexibly anywhere and anytime virtually [15]. In addition to their profitable way of working, these two programs can also be further developed to become new research. Therefore, the program developer uploads the program on Github so that other people can also play a role in developing it further. This is done because Github is a disciplinary site for sharing documents where all people and communities can work together to develop new knowledge [16].

Research by Sastrawangsa et al [10] utilizing the Telegram Bot connected to the server as a provider of document databases for automating student information and services. Rianto et al [9] trying to take a different approach by integrating academic information system services with telegram bots. Syarifudin Zuhri et al's research [17] make it easier for users to automate transactions using telegram bots. The results of this research will be in the form of a bot program to produce a solution for integrated compact automatic mobile penetration tester that never existed before with low resources.

The Telegram bot functions as a liaison between the user (client) and the server providing penetration test tools. That way users can perform remote penetration tests via Telegram as a platform to call the tools found on the bot server. The development of this Telegram bot uses the System Development Life Cycle - Waterfall method where the development is carried out in stages from needs analysis, model making, implementation, testing [18]. As a form of validation of the research, a test was conducted with student respondents from the National Cyber and Crypto Polytechnic or PoltekSSN. This is based on the fact that PoltekSSN is an official university that aims to form human resources in the field of coding and information security [19].

2. Research Methods

This application is made using a method by prioritizing the output or features of a remote-based penetration tester attached to functional and non-functional requirements in Table 2 and Table 3. Remote Method Invocation (RMI) method [20] used to connect telegram bot using remote procedure calls (RPCs) made over HTTP [21] with a Linux server used for the main penetration tester tool. Some of the code is deployed with modifications from the existing github repository as shown in Table 1.

Deployment is done with SDLC waterfall approach. The flow of this deployment can be seen as conceptual framework in Figure 1. The SDLC Waterfall method is very suitable for making new programs or software and can maintain quality because the manufacturing process is carried out in phases [22].

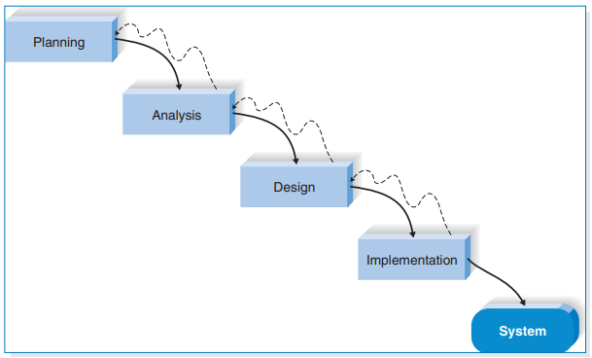

Figure 1. Conceptual framework

2.1 Analysis

The analysis phase is the initial stage carried out to find out what needs are needed in carrying out the design. At this stage, a literature study is carried out to meet the knowledge of the research to be carried out. The results of the literature study are used to formulate problems and determine what needs are needed in designing Telegram bots. In this case what is needed is functional requirements as the main requirements of the system and non-functional requirements as supporting requirements. In addition to some related research, there is also a related github repository used in the development of this application, which can be seen in Table 1.

Table 1. Related Github Repository

Developer	Product	Relevance
Frank Wank	pyTelegram BotAPI	Basic program to build Telegram bot service, create autoresponse, menu, and command list.
Sami Yessou	Remote-shell Telegram	Program to provide Linux terminal services via Telegram bot, contains common command features such as /ping, /dig, and /nmap, and so on.
Harry Suryapamba gya	TelegramB otForPentes ting	Telegram bot program that provides enumeration services in the form of DNS scanner, NMAP scanner, and TCP port scanner
Abhisek Kumar	SQLNUKE	Telegram bot program to provide exploitation tool services in the form of SQL injection

Functional requirements are requirements related to the system and the information contained therein [23]. Table 2 explains that bots have several functional parts, namely the server where the bot runs, Python programming language to build bot programs, penetration test tools as a service feature provided and

DOI: https://doi.org/10.29207/resti.v7ix.xxx

a Telegram bot account as a place where the service can be accessed by users.

Table 2. Functional requirements

No	Functional requirements
1	Linux server as service provider
2	Python programming language for programming bots
3	Penetration testing tools as a service feature
4	Telegram bot account as a place where services can be accessed

Non-functional requirements are requirements that support the system from outside the main requirements. The goal is that the system can serve better. Non-functional requirements can be seen in Table 3.

Table 3. Non-Functional requirements

No	NonFunctional requirements
1	All features of penetration testing tools are integrated into one Telegram bot account
2	The user interface is more friendly with the provision of a menu display
3	There is user authentication by the administrator for security of use

2.2 Design

The design phase includes the design of the program to be created. This design includes the methods used in conducting research, determining the content, function, and workings of the program against bots. In this phase, preparation of the architecture to be built is also carried out, namely by installing the server, installing the tools to be used, and making Telegram bots. Modeling at the program design stage can be described using UML (Unified Modeling Language) which includes use case diagrams, activity diagrams, and sequence diagrams [22].

UML is used to make it easier when designing the design so that every detail of the components of the bot program can be seen clearly. Telegram bot has several functions that can be accessed by users and administrators. The functions of the bot include authentication functions, menu functions, and penetration test tools functions. The use case diagram of the bot is shown in Figure 2.

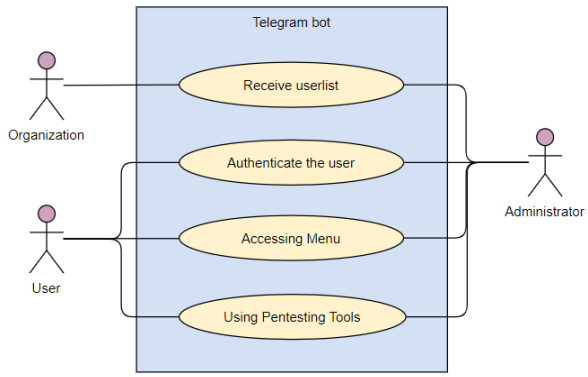

Figure 2. Use Case Diagram

In the use case diagram Figure 2 there are three actors involved in how this Telegram bot works. The organization is the party that wants to apply for permission to use bots to the administrator. Users come from organizations that have registered their members to the administrator in the form of submitting a list of names from users.

On the other hand, the admin is in charge of ensuring the use of Telegram bots from users, especially ensuring that the added users match the list provided by the organization. Because licensing for the use of bots from organizations is manual without going through the Telegram bot system, there are only three main functions that apply to activity diagrams and sequence diagrams. The explanation of the use case diagram can be seen in Table 4 user registration use case, Table 5 authentication use case, Table 6 menu use case and Table 7 penetration test use case.

Table 4. user registration use case

Use case : user registration	
Actor	organization, administrator
description	This use case allows administrators to receive a list of users from an organization who applied for permission to use Telegram bots
Initial conditions	Administrator belum memiliki daftar pengguna bot Telegram
Initial scenario	1. The actor contacts the administrator to ask for permission to use bots in his organization 2. Administrator approves and requests user list 3. The organization provides a list of users who will 4. use bot to administrator
Alternative scenario	-
Final condition	Administrators get a list of users

Table 5. authentication use case

Use case : authentication	
Actor	user, administrator
description	This use case allows actors to be added to the list of users allowed by the administrator so that they can use Telegram bots
Initial conditions	The actor does not have permission to use bots
Initial scenario	1. Actors are looking for Telegram bots 2. Actor running bot with start button 3. The system sends information account to the administrator 4. The administrator gets a message that a new user wants to use the bot 5. Administrator adds a new user account to the list of allowed users 6. The system sends a message to the actor that his account has been allowed to access the Telegram bot
Alternative scenario	The system displays a warning message that the actor's telegram account has not been allowed to use the Telegram bot.
Final condition	The actor's telegram account is on the list of those who have been allowed to use Telegram bots.

DOI: https://doi.org/10.29207/resti.v7ix.xxx

Table 6. menu use case

Use case : menu	
Actor	user
description	This use case allows actors to view the help menu to see instructions for using each of the available tools
Initial conditions	Aktor belum mengetahui bagaimana cara menggunakan bot Telegram
Initial scenario	1. The actor presses the menu button 2. The system displays usage assistance options 3. Actors choose help 4. The system sends how to use the bot along with the available commands from the tools to be selected
Alternative scenario	Actor doesn't access menu feature for help
Final condition	Actors get a message on how to use the tools provided by the Telegram bot

Table 7. penetration test use case

Use case : penetration test	
Actor	user
description	This use case allows actors to use the tools provided by the Telegram bot by sending execution commands
Initial conditions	The actor doesn't have a flag on the challenges page yet
Initial scenario	1. The actor inputs the command 2. The system sends a command to the server to execute 3. The system sends the commands sent by the actor along with the execution results to the actor 4. The system displays that the process has been completed by selecting to send a finish message
Alternative scenario	The actor only gets a finish message from the system without any execution results
Final condition	The actor gets the result of an already executed command

2.2.1 Authentication Function

The authentication function is part of Telegram bot security so bots cannot be abused by unauthorized users. The authentication function can ensure that only users who have been verified by the administrator can use this bot.

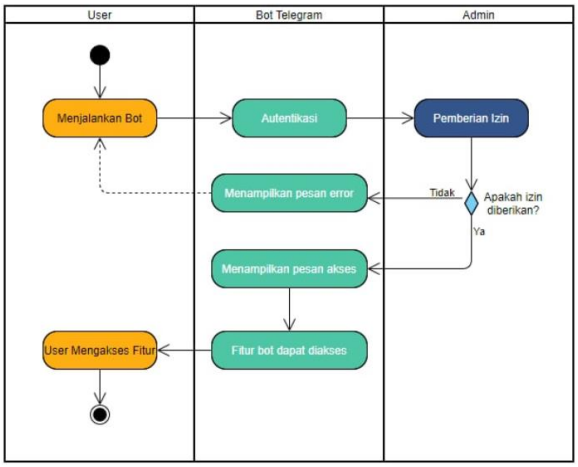

Figure 3. Authentication function activity diagram

An administrator has the right to add new users to the allowed database. The activity flow of the authentication function can be seen in the activity diagram in Figure 3 and the sequence diagram in Figure 4.

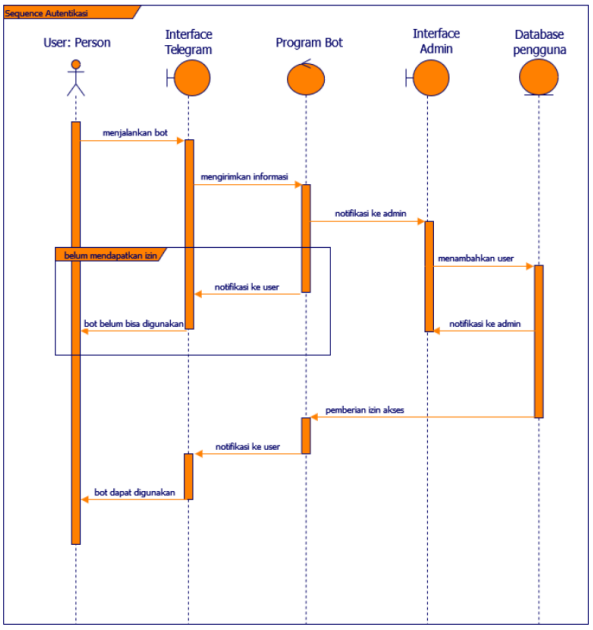

Figure 4. Authentication sequence diagram

2.2.2 Menu Function

The menu function is a function to assist users in providing directions for using Telegram bots. The menu can be accessed by users to display what is provided on the Telegram bot. The menu on the Telegram bot is intended to display examples of using bots, starting from how to input commands to displaying what commands are contained in each of the penetration testing tools that can be performed.

The activity flow of the menu function can be seen in the activity diagram in Figure 5 and the sequence diagram in Figure 6.

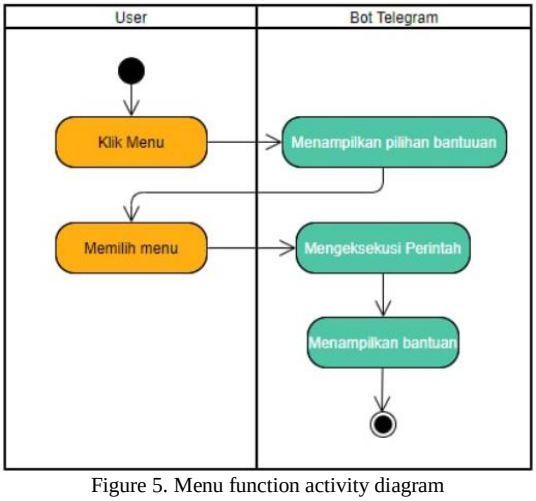

Figure 5. Menu function activity diagram

DOI: https://doi.org/10.29207/resti.v7ix.xxx

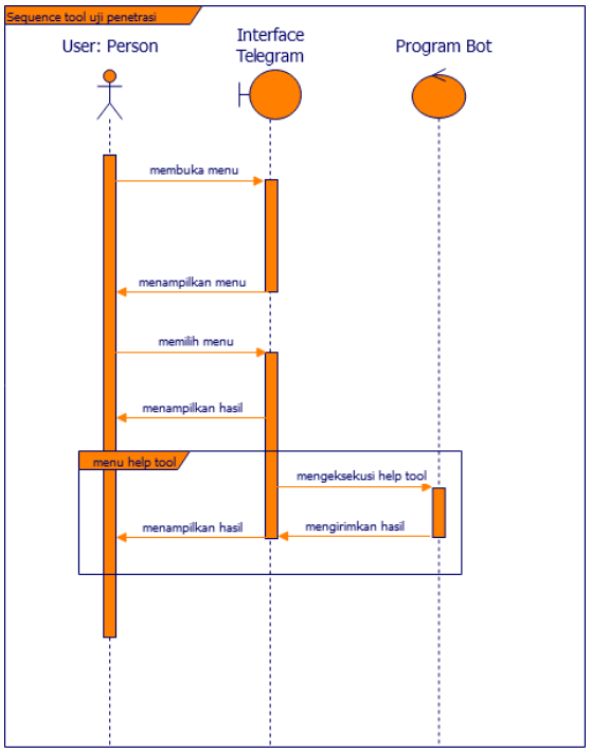

Figure 6. Menu sequence diagrams

2.2.3 Penetration Test Tool Function

The penetration test tool function is the main part of the bot which functions to provide penetration test facilities based on the available tools.

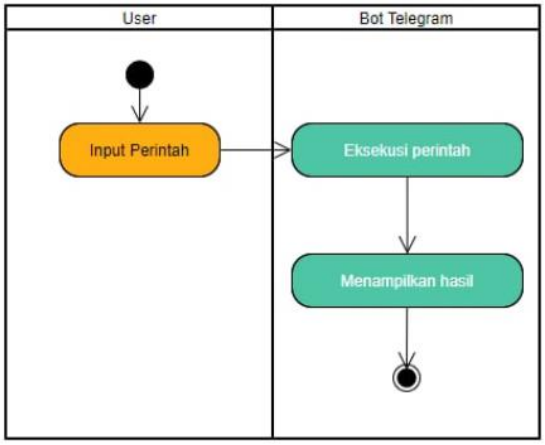

Figure 7. Diagram of activity function of penetration test tools

This function works by executing commands sent by the user on the Telegram bot to be processed on the server. The results of this process will later be sent back to the user via the Telegram bot. The activity flow of the authentication function can be seen in the activity diagram in Figure 7 and the sequence diagram in Figure 8.

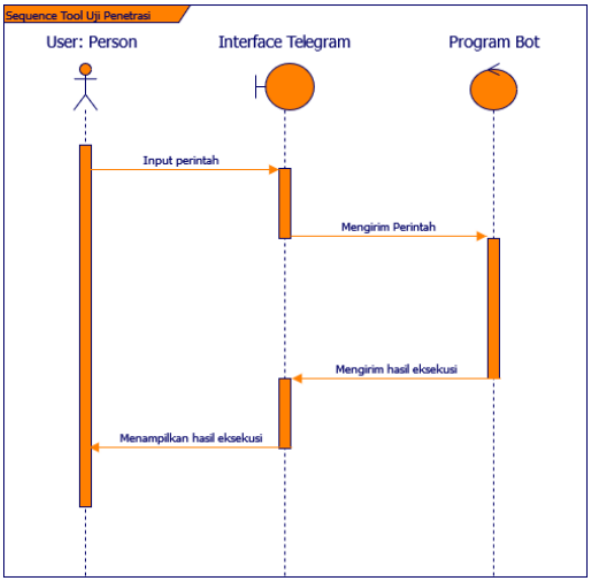

Figure 8. Sequence diagram of penetration test tools

2.3. Implementation

Implementation is a technical phase to apply what is already in the design into the form of a program. This stage produces output in the form of a program that is ready to be used and a Telegram bot that will be connected to the server.

The object of this research is a Remote penetration testing program that provides services to perform penetration tests through Telegram bots. This service is provided by a Linux server that has penetration testing tools installed in it. This means that the Remote penetration testing program connects all existing penetration testing tools on the Linux server to be called via the Telegram bot. The penetration test tools that will be used in this study can be seen in Table 8.

Table 8. Pentesting Tools

No	Tools	Phase	Utility
1.	Subfinder	Information Gathering	Subdomain Enumeration
2.	Nmap	Port Scanning	Port Scanner
3.	Nikto	Vulnerability Scanning	Web Server Scanner
4.	Gobuster	Vulnerability Scanning and Exploitation	Web Server Directory Scanner
5.	Nuclei	Exploitation	Proofing Exploit
6.	Sqlmap	Exploitation	Detecting and Exploiting SQL
7.	Hydra	Password Cracking	Password Cracker

2.4 Testing

The testing phase is carried out by reviewing the results of the programs that have been implemented. The goal is to ensure that the program created can work according to the existing functions in the use case and to test the capacity of the existing architecture. Tests carried out are Black box testing and Load Testing. This

DOI: https://doi.org/10.29207/resti.v7ix.xxx

test refers to the scenario that has been made. The test scenario is the flow of the tests carried out on the Telegram bot program.

2.4.1 Fulfillment of Needs

Fulfillment of needs is required to evaluate Telegram bots based on existing criteria. The criteria needed for this Telegram bot can be seen in Table 9.

Table 9. Kriteria pemenuhan kebutuhan

No	Needs	Characteristics
1	Telegram bots have easy access.	Scenario (Accessibility)
2	The use of the Telegram bot is easy and understandable.	Scenario (Use)
3	Telegram bot provides adequate penetration testing tools.	Scenario (Tools)
4	Telegram bots are more practical to use than going through the operating system directly.	Scenario (Resource)
5	Telegram bots can meet the need for penetration testing.	Scenario (Requirement)

2.4.2. Questionnaire

assessment of this questionnaire using Likert scale of 5 indicates Strongly Agree (SS), scale 4 Agree (S), scale (3) Undecided (R), scale 2 Disagree (TS), or scale 1 Strongly Disagree (STS). Questionnaire made with a Likert scale and analyzed using the Formula 1 [24], [25].

$$\frac{\sum(a \times b)}{c \times d} x100\% \quad (1)$$

a is the total number of respondents who chose the answer b, b is the answer choice score, c is the total number of respondents, and d is the highest score.

To find out the level of acceptance, the calculation results in the previous formula are then added up and recalculated using Formula 2 [24], [25].

$$\frac{\sum w}{x \times y \times z} x100\% \quad (2)$$

w is the total score of the respondent's choice, x is the total statements, y is the total respondents, and z is the highest score.

2.4.3. Respondent Selection

The selected respondents were students of the Cyber and Crypto Polytechnic which were part of the Senate organization of Cadets Corps. This is based on the fact that cyber security is studied by students so that it can be used for learning penetration tests.

3. Results and Discussions

3.1 Telegram Bot

Telegram bot in Figure 9 was developed using the Python 3 language as can be seen and begins with the authentication function, the source code snippet is shown in Figure 10 with the output authentication function in Figure 11.

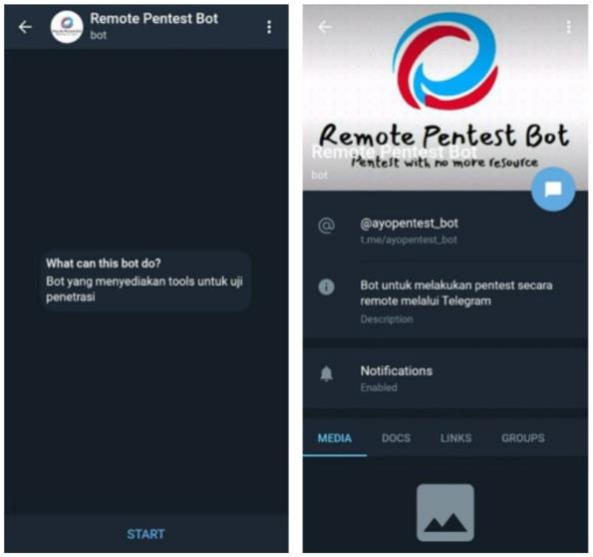

Figure 9. Pentest telegram bot

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os
import time
import checkword
import telebot
from telebot import types
from telebot import util
from telebot.types import ReplyKeyboardMarkup, ReplyKeyboardRemove,
KeyboardButton

from config import TOKEN

checkword.add good words(['nikto','nmap','gobuster','sqlmap','hydra','n
uclei','subfinder'])

file =  open('admins.txt',"r")
admins = []
for val in file.read().split():
    admins.append(int(val))
file.close()

command_error = "Error! Harap masukkan command yang valid, daftar
command dapat dilihat pada /help"
request = "Permintaan anda telah dikirimkan ke admin, notifikasi akan
dikirimkan ketika sudah diizinkan."
admin_approval_form = "Terdapat permintaan baru untuk menggunakan bot"

# AUTH
def validating(cid):
    file = open('allowed user.txt', "r")
    allowed_user = []
    for val in file.read().split():
        allowed_user.append(int(val))
    file.close()
    if cid in allowed user:
        return True
    else:
        return False

def failed_attempts(cid,name,content):
    with open('failed_attempts.txt',"a") as file:
        file.write(f"""{cid} {name} - {content}\n""")
```

Figure 10. Authentication Source Code

After the user is authenticated, the user can use the bot with the menu in capture code Figure 12 followed by the output in Figure 13.

In the function section of the penetration test tool in Figure 14, code is written to set the whitelist of allowed words, the use of the "/exec" command so that users do not need to run the tools by opening them one by one, and displaying the status of the ongoing process by sending "run" and "running" messages. Process Completed" when finished. Writing code for penetration test tools can be seen in Figure 15.

DOI: https://doi.org/10.29207/resti.v7ix.xxx

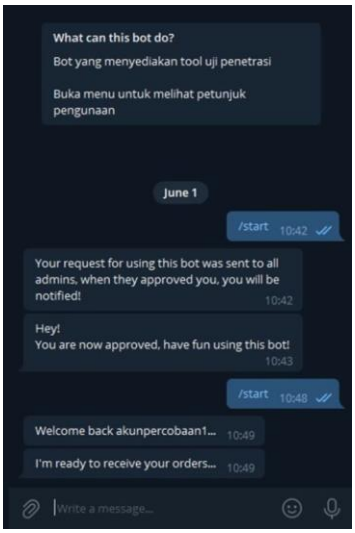

Figure 11. Authentication Function

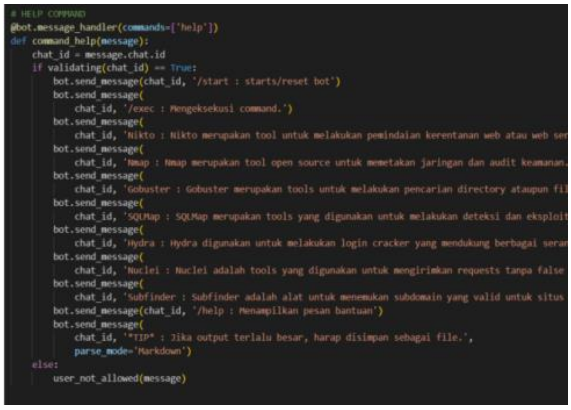

Figure 12. Menu code function

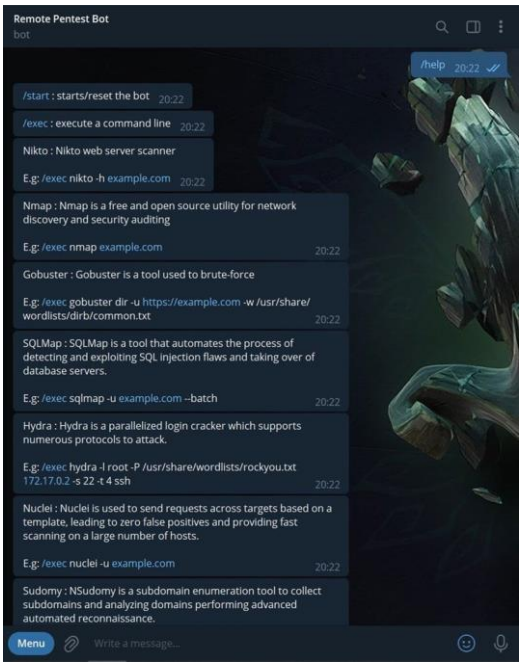

Figure 13. Menu function

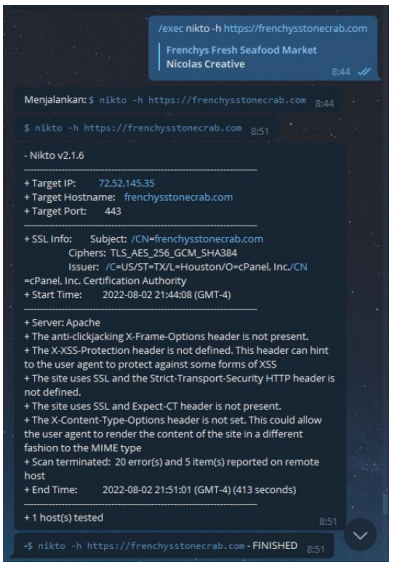

Figure 14. Pentesting Tools Function

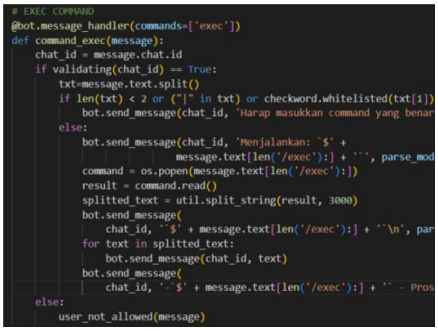

Figure 15. Pentesting Tools code Function

3.2 Server

The server on this Telegram bot uses a server which must be activated first to run properly. The specifications of the server in this case is virtual machine can be seen in Figure 16. On the server there are several files in one folder which are part of the bot program unit. These files and their functions can be seen in Table 10. To provide services to Telegram bots, the bot program must be executed on the terminal server first. That way the bot can respond to user requests when used by the user.

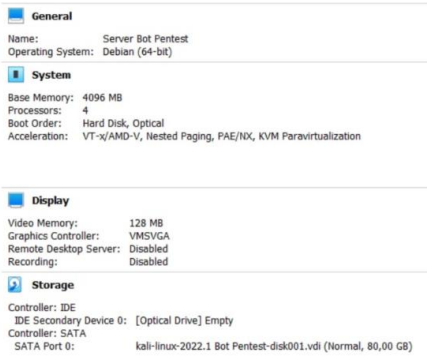

Figure 16. server spesification

DOI: https://doi.org/10.29207/resti.v7ix.xxx

Table 10. File output in server

No	Filename	Fuction
1	_pycache_	Cache folder from bot program use
2	admins.txt	List of accounts registered as administrators
3	allowed_user. txt	List of accounts allowed by the administrator to access the Telegram bot
4	config.py	Configuration between the Telegram bot and the bot program on the Kali Linux Server
5	failed_attempts .txt	List of failed attempts to access Telegram bots because they have not been authorized by the administrator
6	logFileBot. txt	Bot usage logs stored while the server is up
7	pentest_bot. py	The main program of the Telegram bot to provide services on the user interface in Telegram

3.2 Testing

This test involves three entities, namely the Senate Organization of the PoltekSSN Youth Corps, students, and administrators. The Senate Organization of the PoltekSSN Taruna Corps plays a role in registering its members and providing a user list containing the names and classes of students. Students play a role in accessing and using Telegram bots. At the time of trying to access, the administrator will be able to see the username and chat id of the student. Meanwhile, the Administrator plays the role of receiving data from the Organization and authenticating if there is a data match. Administrators can update the list of allowed users to the server.

3.2.1 Black Box Testing

Testing using black box testing is used to determine test results based on program functionality. Black box testing was chosen because it only focuses on testing the functional requirements based on the program created [26]. Therefore this test is also called functional testing. By using functional testing, the ability of the system to interact with inputs can be assessed [27]. The test output can be seen in Table 11.

Table 11. Black Box Testing result

Function	Testing scenario	input	Expected output	Results	Test conclusion
authentication	Open Telegram bot	New users press the start button, Admin press the add user button	Telegram bot sends information to admins and new users that a new user account has been added	Admin gets message new user has been added from Telegram bot, new user gets message his account has been added from Telegram bot	Succeed
menu	Open Telegram bot	Press the menu button, select the help menuPress the menu button, select the help menu	The menu is displayed, the bot executes the help menu and sends the results to the user	Users can select the help menu and get help information sent by the Telegram bot	Succeed
Penetration test tool	Open Telegram bot	Type "/exec" then enter the command according to the selected tool and tool features	Telegram bot sends back user inputted commands, Telegram bot sends command execution results, Telegram bot sends a message the process is complete	The user gets a command message that has been entered, the user gets information on the results of the execution, the user gets a message the process is complete	Succeed

3.2.2 Load Testing

Load testing is carried out to determine the capacity of the bot program to determine the extent to which the program can accommodate a large number of loads [28]. Load testing is done by activating the Monitorix tools [29] on the on server within one hour. This tool is used when server activity is high because it serves the users. The monitoring results from the system load recorded by Monitorix can be seen in Figure 17. From the test, it was found that the highest load average value was 5.4, which contained processes running for an average time of one minute with a total of 35 users.

3.2.3 User Acceptance Test

The UAT assessment is taken based on a questionnaire that has been filled out by a number of users. To determine the level of acceptance, the UAT calculation formula is used. This formula calculates the percentage by dividing the score of the questionnaire which is worth 785 by 5 statements, 35 respondents, and the highest score is 5 points. The score is in the "strongly agree" acceptance area shown in Table 12 and Figure 18 .

DOI: https://doi.org/10.29207/resti.v7ix.xxx

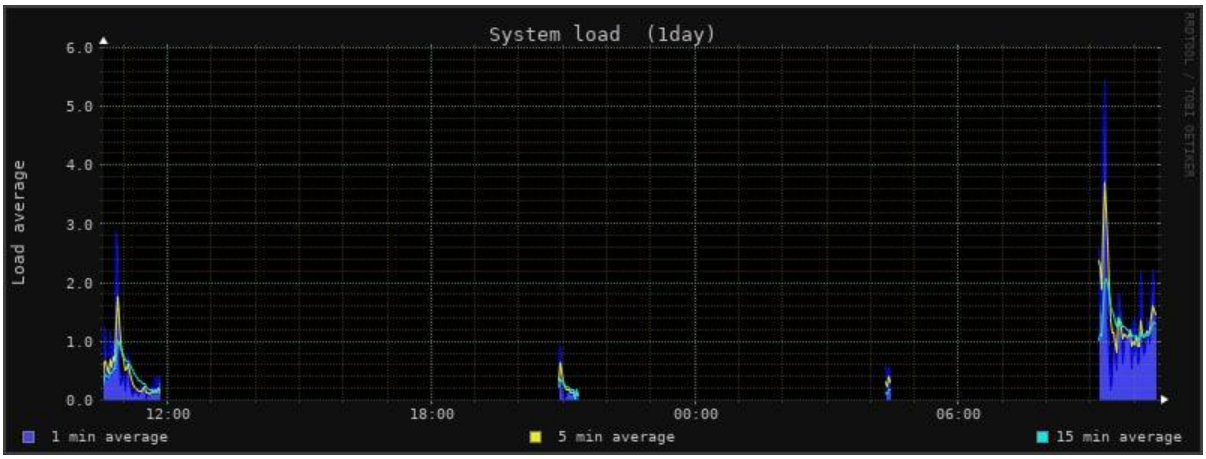

Figure 17. Load Testing

Table 12. Response User Acceptance Test

respondent	Question				
	1	2	3	4	5
1	5	5	4	4	4
2	5	5	4	5	4
3	5	5	5	5	5
4	5	5	5	5	3
5	4	5	5	5	5
6	4	3	5	5	4
7	5	5	5	5	5
8	5	5	2	4	2
9	5	5	4	4	4
10	5	5	5	5	5
11	5	5	5	5	4
12	5	5	5	5	5
13	5	5	4	5	4
14	4	4	4	4	4
15	5	5	5	5	5
16	4	4	4	4	3
17	5	5	5	5	5
18	5	5	4	5	4
19	5	4	4	5	3
20	5	5	4	4	4
21	4	4	4	4	4
22	5	5	4	4	4
23	5	5	5	5	5
24	4	4	4	3	3
25	5	3	4	4	4
26	4	4	4	4	4
27	5	5	4	5	4
28	4	4	4	4	4
29	5	4	3	4	4
30	4	4	4	4	4
31	5	5	5	5	5
32	4	4	3	4	3
33	5	5	5	5	5
34	5	5	3	5	3
35	4	4	4	5	5
Total	164	160	148	159	143
Percentage	93,71%	91,42%	84,57%	90,85%	81,71%

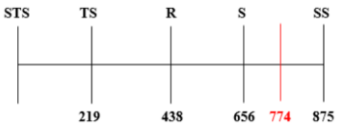

Figure 18. UAT assessment areas

4. Conclusion

Telegram bot is able to integrate well and fulfill all functional and non-functional requirements. Based on load testing, the maximum limit of users who can access Telegram bots simultaneously is 35 users with the highest load average of 5.4. Based on the results of the User Acceptance Test, the Telegram bot has an acceptance rate score of 88,457% and a questionnaire score of 774 which is an agreed area. In other words, this telegram bot is a solution for integrated compact automatic mobile penetration tester with low resources. For future research this mobile penetration tester can be equipped with auto complete command features, command error warnings and the addition of live output from the server so that it does not only display text-based output.

Acknowledgment

This research was partially supported by National Cyber and Crypto Polytechnic.

Reference

[1] G. Werner, S. Yang, and K. McConky, "Time series forecasting of cyber attack intensity," *ACM Int. Conf. Proceeding Ser.*, 2017,

[2] I. B. M. Security, "IBM: 2021 X-Force Threat Intelligence Index," 2021.

[3] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb, and S. Mahmood, "Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study," *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 3171–3189, 2020,

[4] G. Jayasuryapal, P. M. Pranay, H. Kaur, and Swati, "A Survey on Network Penetration Testing," *Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021*, pp. 373–378, 2021,

[5] H. M. Z. Al Shebli and B. D. Beheshti, "A study on penetration testing process and tools," *2018 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2018*, pp. 1–7, 2018,

[6] N. Bhingardeve and S. Franklin, "A Comparison Study of Open Source Penetration Testing Tools," *Int. J. Trend Sci. Res. Dev. (IJTSRD)*, pp. 2595–2597, 2018.

[7] A. D. Nobari, N. Reshadatmand, and M. Neshati, "Analysis of telegram, an instant messaging service," *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. Part F1318, pp. 2035–2038, 2017,

[8] W. K. Pertiwi, "Telegram Genap Berusia 8 Tahun, Apa Saja Pencapaiannya?," *Kompas.com*. 2021. [Online]. Available: https://tekno.kompas.com/read/2021/08/16/17020017/telegram-genap-berusia-8-tahun-apa-saja-pencapaiannya?page=all

[9] R. Rianto, A. Rahmatulloh, and T. A. Firmansah, "Telegram

DOI: https://doi.org/10.29207/resti.v7ix.xxx

Bot Implementation in Academic Information Services with The Forward Chaining Method," *Sinkron*, vol. 3, no. 2, pp. 73–78, 2019,

[10] G. Sastrawangsa, "Pemanfaatan Telegram Bot Untuk Automatisasi Layanan Dan Informasi Mahasiswa Dalam Konsep Smart Campus," *Konf. Nas. Sist. Inform.*, p. 773, 2017.

[11] S. Yessou, "Telegram Remote-Shell," *Github*. 2020. [Online]. Available: https://github.com/fnzv/trsh

[12] H. Suryapambagya, "Telegram bot for pentest," *Github*. 2018. [Online]. Available: https://github.com/AmikomVirusCommunity/telegram-bot-for-pentest

[13] A. Muharam, "Kerja Remote, Tren Sistem Kerja Saat Ini." 2018. [Online]. Available: https://www.logique.co.id/blog/2018/04/27/kerja-remote-tren-sistem-kerja-saat-ini/

[14] "State of Remote Work," *Owl Labs*. 2020. [Online]. Available: https://owllabs.com/state-of-remote-work/2020

[15] H. W. Lim, "Implementing Remote Working Policy in Corporate Offices in Thailand : Strategic Facility Management Perspective," 2021.

[16] D. R. Bouqin, "Github," *J. Med. Libr. Assoc.*, vol. 103, no. 3, pp. 1667–1668, 2015.

[17] S. Zuhri, G. I. Marthasari, and Y. Azhar, "Otomatisasi Transaksi Toko Online Berbasis Woocommerce Menggunakan Bot Telegram," *J. Repos.*, vol. 2, no. 6, p. 717, 2020,

[18] M. Ridwan, I. Fitri, and B. Benrahman, "Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 5, no. 2, p. 173, 2021,

[19] "Politeknik Siber dan Sandi Negara." [Online]. Available: https://poltekssn.ac.id/

[20] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. Elsevier Science, 2011. [Online]. Available: https://books.google.co.id/books?id=BvaFreun1W8C

[21] J. Yi, "Introduction to the Telegram Bot API, Part 1 Chatbots Life," 2019. https://chatbotslife.com/introduction-to-the-telegram-bot-api-part-1-2ae36f7b30a4 (accessed Apr. 30, 2023).

[22] R. Bahar; Wibawa, Basuki; Situmorang, *Rekayasa Perangkat Lunak: Pendekatan Terstruktur & Berorientasi Objek*.

[23] R. S, *System Analysis and Design*, 7th editio., vol. 02, no. 05. Wiley Publishing, Inc, 2012.

[24] S. Chakrabartty and S. Nath Chakrabartty, "Scoring and analysis of likert scale: Few approaches," *J. Knowl. Manag. Inf. Technol.*, vol. 1, no. 2, pp. 31–44, 2019, [Online]. Available: https://www.researchgate.net/publication/321268871

[25] A. Joshi, S. Kale, S. Chandel, and D. Pal, "Likert Scale: Explored and Explained," *Br. J. Appl. Sci. Technol.*, vol. 7, no. 4, pp. 396–403, 2015,

[26] T. Snadhika Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *J. Inform. J. Pengemb. IT*, vol. 03, no. 02, pp. 45–48, 2018,

[27] D. Fatiyah, A. Chusnul; Gumilang, S.F Surya; Witarsyah, "Pengujian Fungsional Dan Non Fungsional Aplikasi Web Borongajayu," vol. 6, no. 2, pp. 2–8, 2019.

[28] Z. M. Jiang and A. E. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," *IEEE Trans. Softw. Eng.*, vol. 41, no. 11, pp. 1091–1118, Nov. 2015,

[29] I. Free Software Foundation, "Monitorix." Accessed: Jul. 20, 2022. [Online]. Available: https://www.monitorix.org/

DOI: https://doi.org/10.29207/resti.v7ix.xxx