Research Article

International Journal of Information System and Technology (IJOINT)
Volume 01, Issue 03, PP. 104–114
E.ISSN: 3090-5672

Penetration Testing System CERDAS With Brute Force Method

Ronald Merlang[1*], and Apri Siswanto[3]

[1] Student, Informatics Engineering Study Program, Riau Islamic University Pekanbaru, Indonesia
Email: rmerlang@gmail.com [1]
[2] Lecture, Informatics Engineering Study Program, Riau Islamic University Pekanbaru, Indonesia
Email: aprisiswanto@eng.uir.ac.id [2]

*Corresponding Author Email: rmerlang@gmail.com

Received: April 17, 2025; Revised: Mei 01, 2025; Accepted: Mei 23, 2025

Abstract

Security of access rights is an important aspect so that there is no misuse by individuals seeking profit. The national cybersecurity operations center and the state cyber code agency recorded 88 million cyber-attacks in 2020. In an effort to prevent attacks on CERDAS, the author conducted a study by raising the problem "Can the security system used by CERDAS prevent brute force attacks and how is the brute force attack process carried out". The purpose of this study was to test the login page security system used by CERDAS. The method used was NIST 800-115. The results of the study showed that the login page security system used by CERDAS had a gap against brute force attacks with a success rate of up to 100%. Based on the results of the study, CERDAS must improve the login page security system as an effort to avoid brute force attacks that aim to harm the system.

Keyword: Penetration Testing, Brute Force, Access Security, System CERDAS.

1. Introduction

Information security is an important aspect to prevent misuse of access rights, information and other possible losses that may occur from a system (Alkhudhayr et al., 2019); (Farahmand et al., 2005). One effort that can be made to maintain security is by evaluating the system used (Shah & Mehtre, 2015). The form of evaluation that can be done is a system resilience test (Penetration Testing) (Denis et al., 2016); (Edwards, 2024). The National Cyber Security Operations Center (PUSOPSKAMSINAS) and the National Cyber and Crypto Agency (BSSN) recorded that there were 88 million cyber-attacks that occurred from January to April 2020 with 25 million attacks in January, 29 million in February, 26 million in March and 7.5 million on 1-12 April[1]. The details of the attacks that occurred were 56% trojan activity, 43% information gathering and 1% web application attack.

The Center of E-learning and Education for Students (CERDAS) of the Islamic University of Riau is a system built for the learning needs of students of the Islamic University of Riau. In CERDAS there is some information from students that is personal and must be kept secure. but on the other hand, its security remains vulnerable to various cyber threats (Abomhara & Køien, 2015), one of which is a brute force attack—a method that attempts to penetrate the authentication mechanism by repeatedly testing various combinations of usernames and passwords (Al Sharaa & Thuneibat, 2024); (Sowmya et al., 2012).

[1] https://www.kompas.com/tren/read/2020/04/23/165400665/bssn-catat-adanya-88-4-juta-serangan-siber-selama-pandemi-corona?page=all

To anticipate this risk, penetration testing is an important step to identify system weaknesses before they can be exploited by irresponsible parties (Blakley et al., 2001); (Sari & Pakaja, 2024); (Vishwakarma & Jain, 2020). This study focuses on testing the security of the CERDAS System by implementing the brute force method using Hatch to evaluate the strength of the login mechanism, measure the time required to break into the system, and analyze the factors that influence these vulnerabilities. The results of this test are expected to provide recommendations for improvements to improve system security, such as implementing account lockout, captcha, or multi-factor authentication (MFA), so that the CERDAS System can operate more safely and reliably in the face of cyber threats in the future.

2. The Art of Research

1. Penetration Testing

Penetration Testing is a test to find loopholes or weaknesses in a system by attacking the system with the aim of improving system security in order to minimize attacks from individuals who can harm the owner of the system (Sari & Pakaja, 2024); (Yaqoob et al., 2017). Singh & Sharma (2020) argue that penetration Testing has 6 types, namely:

a. Blind, the Computer Network Tester (Pentester) gets only a little information from the system to be attacked, while the target knows and has prepared the system for the attack.
b. Double-blind, the Computer Network Tester is not provided with sufficient information about the system to be attacked and the target also does not prepare the system for the attack test process.
c. Graybox, the target party has prepared the system for testing while the Pentester is only given a little information from the target group regarding the system to be tested.
d. Double graybox, the target party prepares the system for the penetration testing process and will inform the Computer Network Tester of the scope of the test to be carried out.
e. Tandem, the target party and the Computer Network Tester prepare the system to be tested together.
f. Reversal, the target party does not prepare and does not know the system will be tested while the Pentester knows in detail about the system to be tested. Usually, the pentester and the target party come from the same organization.

2. System CERDAS

Center Of E-Learning and Education for Students (CERDAS) is an online-based application used to meet the needs of online learning or E-Learning for students that can be accessed via gadgets such as smartphones or laptops. CERDAS was launched on March 12, 2021. CERDAS provides several services such as discussion media between students and lecturers, a place to collect assignments, providing information related to lectures and providing student academic history.

3. Brute Force

Brute Force is a form of system attack by trying all combinations to find the right password from a system (Ayankoya & Ohwo, 2019). Brute force attacks are carried out repeatedly and will stop when the password is found, the given combination has run out and when prevented by the security system of a system (Gautam & Jain, 2015). Brute force uses an algorithm to solve problems simply (Alkhwaja et al., 2023). The advantage of brute force is that it works simply but has a good level of effectiveness for some systems. The disadvantage of brute force is that it takes a lot of time to find a password combination.

4. Research Theory Development

Recent research in cybersecurity, such as a study by Ayankoya & Ohwo et al. (2029) published in the International Journal of Computer Science and Information Security, shows that brute force attacks are still

a significant threat to authentication-based systems, with a 35% increase in cases in 2022–2023, especially in systems that do not implement basic protections such as rate limiting or multi-factor authentication (MFA). Findings from Inayat et al. (2026) published in the Journal of Network and Computer Applications also revealed that response systems are often the main target because they store critical data, but many of them still use weak password policies. This study aims to analyze the vulnerability of the System CERDAS to brute force attacks by adapting the penetration testing methodology (Owens & Matthews, 2008), which proves that a combination of tools such as Hydra and John the Ripper can reveal security vulnerabilities 30% faster than conventional approaches. The test results are expected to not only identify system weaknesses but also strengthen current findings with evidence-based recommendations, such as the implementation of adaptive authentication (Mohammadi et al., 2019) or behavioral biometrics algorithms (Del-Valle-Soto et al., 2024), to mitigate the risk of exploitation in intelligent system environments. This study uses hatch, a Python-based brute force tool. Users must prepare a wordlist to run Hatch, and Hatch will attack until the password is found or the provided wordlist runs out.

3. Method

1. Pentest Simulation Plan

Referring to the NIST SP document with code 800-15 penetration testing is divided into several stages, namely planning, discovery, attack and reporting (see figure 1). The CERDAS system penetration testing simulation plan based on the NIST method is as follows:

1) Planning through the development of a CERDAS clone system, carrying out attacks on the CERDAS clone system using the Kali Linux virtual operating system and the Hatch tool, making reports on the penetration testing carried out.

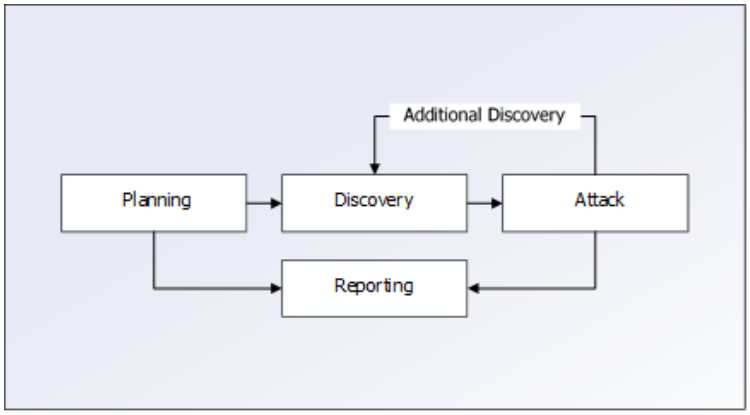

Figure 1. NIST Penetration Testing Methodology

1) Discovery is an effort to identify security gaps in a mock system by conducting observations and searching for literature studies that support this penetration testing.
2) Attack the mock system to verify the security gaps identified in the discovery stage.
3) Reporting and recording the results of penetration testing that has been carried out

2. System Vulnerability Testing Tools

The tools used in the research process consist of hardware and software. The hardware used is a laptop and the software used is XAMPP and Hatch and several other supporting software. Virtual Box to simulate CERDAS, XAMPP as a CERDAS web server and Hatch tool to perform pentest. The specifications of the tools and testing requirements will be presented in table 1 as follows:

Table 1. PENTEST Equipment Specifications and Requirements

Component	Specification	Function
Processor	Intel® Core i5-10300H CPU (2.50 GHz)	Processing instructions given by the application
RAM	8 GB	Temporary storage while the processor processes commands
Storage Memory	500 GB	Application storage location
VGA	NVIDIA GeForce GTX 1650 Ti 4GB GDDR6	Provides output in the form of a display from the application

3. System Vulnerability Testing Tools

   The tools used in the research process consist of hardware and software. The hardware used is a laptop and the software used is XAMPP and Hatch and several other supporting software. Virtual Box to simulate CERDAS, XAMPP as a CERDAS web server and Hatch tool to perform pentest. The specifications of the tools and testing requirements will be presented in table 1 as follows:

4. Application Requirements

   1. Hatch, had using hatch, each user must provide a collection of words (wordlist) that may be the password of the target system. The more we know the system or user habits in determining passwords, the greater the level of success in entering the system by force (Florencio & Herley, 2007). Hatch will carry out repeated attacks until the wordlist runs out or when Hatch finds the right password combination. The workflow of using hatch in this study will be presented in Figure 2 below.

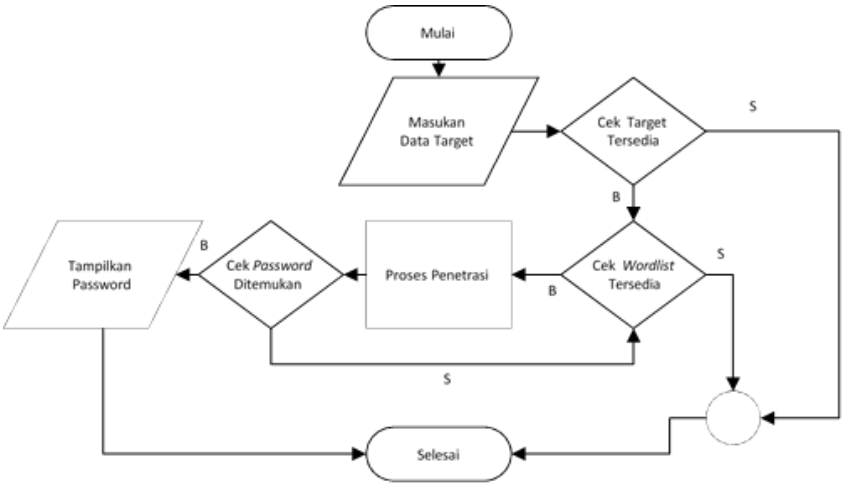

Figure 2. Hatch System Workflow

   2. Design Model CERDAS is adjusted to the actual CERDAS functionality. This model requires a web server as a service provider in order to run and be adjusted for research needs (see figure 3). Next, add username and password data so that it is recorded in the server database (see figure 4 for the flowchart of the CERDAS imitation system logic flow).

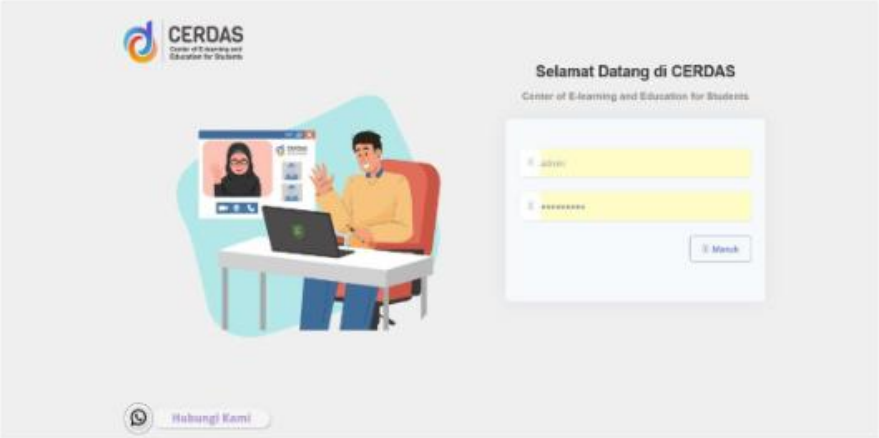

Figure 3. Initial mock-up of the SMART system

If the data entered into this fake system is correct, the user will be able to log in to the system, and if the data entered is incorrect, the user will receive a message that the username or password entered is incorrect.

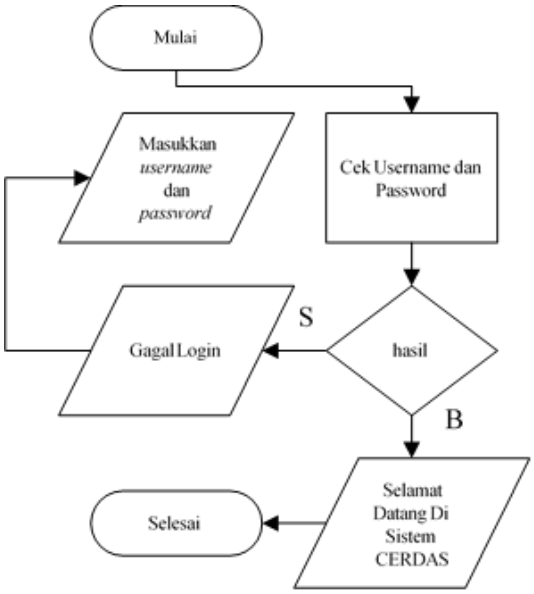

Figure 4. Flowchart of the SMART Artificial System Logic Flow

4. Result

1) Planning

At this stage, the Malukan research team prepares the initial research needs such as the Kali Linux operating system and hatch tools. The installation of the Kali Linux operating system is done virtually using Virtual Box and the installation of the hatch tools (see Figure 5) will be done in the Kali Linux operating system.

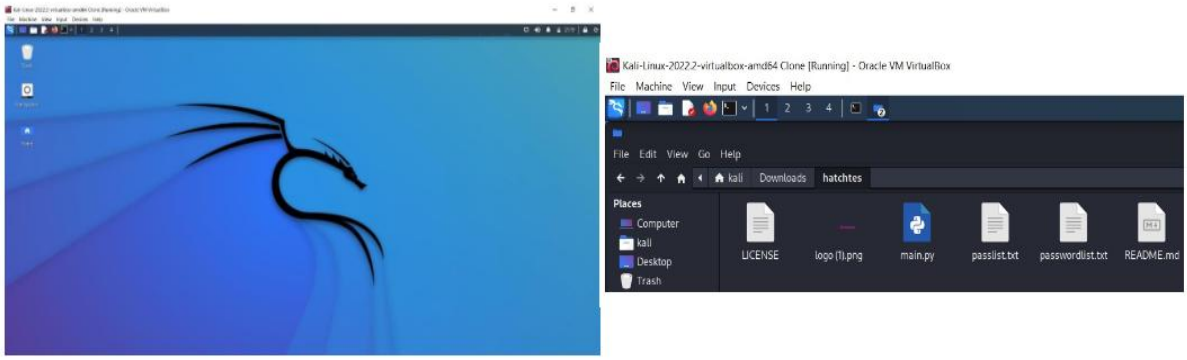

Figure 5. Kali Linux Operating System and Hatch Tools Directory Installation

2) Discovery

After conducting observations and several attempts to enter the CERDAS system, the author obtained information about how to enter the target system. Furthermore, the research team developed a CERDAS system imitation model using PHP language and utilizing XAMPP as a database and web server service provider. The development of this imitation system was made in 2 stages, namely: creating a duplicate system according to the original display on the official CERDAS system page and conducting testing with the blackbox method for the login process that will be carried out by system users. In this case, the results of the simulation and blackbox testing carried out on this imitation system obtained results that were in accordance with the research provisions (see table 2).

Table 2. Results of the Simulation System Test

Test Scenario	Test Case	Expected results	Test Results
Form Username dan Password	Empty the username and password fields Clear the username or password field	Displays the note "Fill Out This Field "	As Expected
	Filling in the Username or password field incorrectly	Displays the message "Login Failed, Username or Password is incorrect"	As Expected

3) Attack

The attack was carried out through Hatch installed on the Kali Linux virtual operating system with the target location being the CERDAS model installed on the main laptop (see figure 6). This test begins by providing a collection of data or wordlists that are commonly used for password searches. The collection of wordlists is divided into 2, namely passwords derived from the user's date of birth because they are the default passwords for logging into the CERDAS system and creating a collection of passwords consisting of common student words and a combination of numbers.

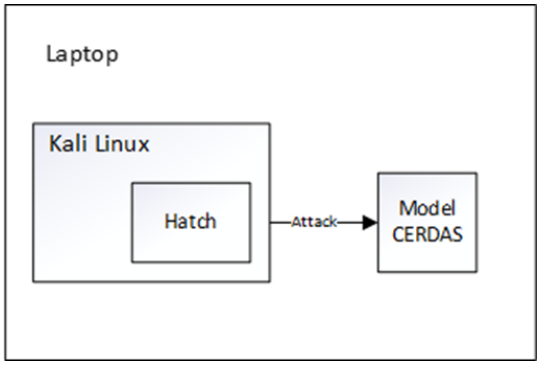

Figure 6. Penetration Testing Scheme

a. Creating a wordlist based on date of birth

The creation of a birth date wordlist uses a python-based tool called date-generator. with the command format "python3 date_generator.py {starting year} {ending year} {display format} {separator}". The creation of the wordlist starts from 1990 to 2005 which is saved with the name passwordlist.txt. General wordlist creation (see figure 7).

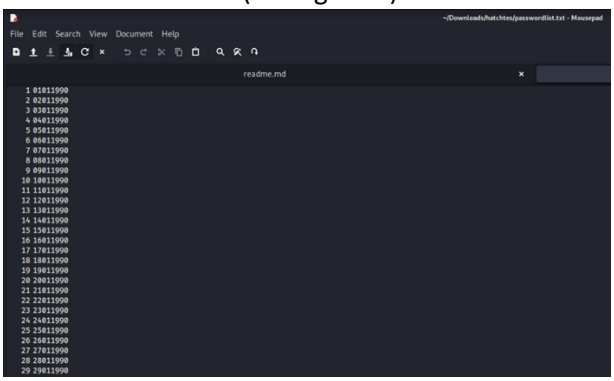

Figure 7. Birth Date Wordlist Collection

b. Creating a general wordlist

In making a general wordlist the author took from the page "https://github.com/geovedi/indonesian-wordlist/blob/master/05-ivanlanin2011-sort-alpha.lst" which provides a collection of wordlists totaling 18313 wordlist lines (see figure 8). The wordlist collection is combined in one file called passwordlist.txt so that the total wordlist collection is 24308 lines.

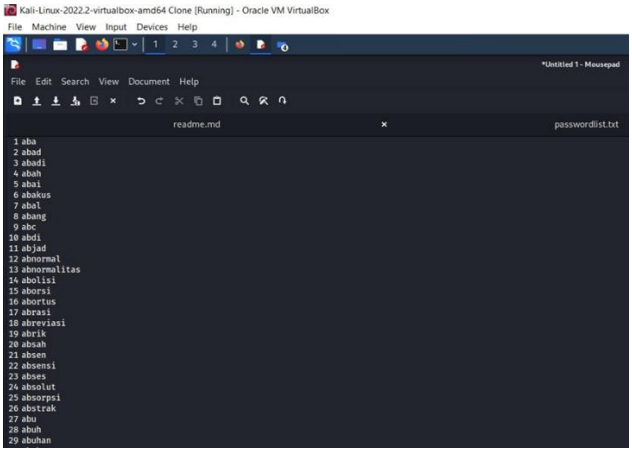

Figure 8. General Wordlist Collection

Next, the total target to be attacked is those who are registered users with username and password that have been given input data in such a way with the provisions of the date of birth and a collection of random words. The initial attack process is carried out through checking the target to be attacked using Hatch (see figure 9) and in this process hatch will carry out the attack technique with the CSS form login selector method (see figure 10).

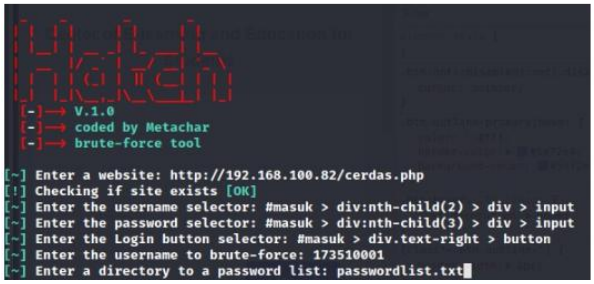

Figure 9. Checking the Target Wordlist Using Hatch

Hatch will provide results which are the number of attack attempts, the password used by the user and the total time spent on the attack and if Hatch finds a password then the tool will stop immediately.

Figure 10. Hack Attack Technique Using CSS Login Form Selector

4) Reporting

In the reporting stage, the results of the attacks that have been carried out are recorded and made in the form of separate tables between the attempted attacks on the default password and the password that has been changed. After conducting several attempts, the hatch attack showed several using the provisions of the user password that still used the initial provisions, namely the date of birth (see table 3) and the results of the attack trials showed 100% success in finding the password in these provisions.

Table 3. Results of Username Attack with Initial Conditions

No	Username	Total Wordlist	Time/ Second	Password Found
1	173510000	3261	1041	√
2	173510002	1895	814	√
3	173510003	1624	711	√
4	193510001	4256	1828	√
5	183510787	3916	1993	√
6	203510021	3639	1970	√
7	203510566	4563	2104	√
8	203510723	4871	1712	√
9	181032001	3606	1439	√
10	191103099	3883	1545	√

Furthermore, the results of the attack on users who had changed their passwords showed a 30% success rate for the attempted attack (see table 4).

Table 4. Results of Attacks on Usernames Who Have Changed Passwords

No	Username	Total Wordlist	Time/ Second	Password Found
1	173510001	24308	10583	x
2	203510001	5954	2489	√
3	213510007	5964	2590	√
4	203510002	6063	3419	√
5	219978033	24308	9196	x
6	209873105	24308	12500	x
7	167809998	24308	11974	x
8	175789801	24308	10877	x
9	191078567	24308	11378	x
10	207219876	24308	12577	x

5. Discussion

The final results of the two experiments, if combined, the success rate for finding the password was 13 out of 20 attempts with a success rate of 65% and the success rate for implementing a brute force attack using Hatch was 20 out of 20 attempts, which is 100%. Based on the results of the attack, it can be concluded that the group of users who have changed their passwords have a better level of security, because the brute force dictionary attack uses passwords that are often used by users in general, so the more complex the password used, the more difficult it is for the password to be found.

Brute force dictionary attack is a cyber-attack method that aims to guess or crack a password by trying various combinations of commonly used words or phrases, usually from a dictionary that contains a list of popular passwords, common phrases, or words that users often use. Unlike a pure brute force attack that tries all possible characters randomly, a dictionary attack is more targeted and efficient because it takes advantage of the human tendency to use easy-to-remember passwords, such as names, birth dates, or common words such as "password123" or "admin". Attackers typically use predefined wordlists—such as RockYou.txt (containing millions of leaked passwords from data breaches) or SecLists—integrated with tools such as Hydra, John the Ripper, or Hashcat to automate login attempts. The speed of these attacks depends on the length and complexity of the password, as well as the presence of security mechanisms such as account lockout, rate limiting, or CAPTCHA. If the system does not have such protection, these attacks can succeed in minutes, especially if the user's password is weak. Research by Verizon (2023) in the Data Breach Investigations Report states that 80% of breaches related to credential stuffing and brute force are successful because of the use of easy-to-guess passwords. Therefore, mitigations such as implementing multi-factor authentication (MFA), complex password policies (minimum 12 characters with a combination of upper/lowercase letters, numbers, and symbols), and monitoring failed login attempts are crucial to prevent this exploitation.

6. Conclusion

The results of the penetration testing simulation on the CERDAS system with the brute force method using hatch show that there is a vulnerability to brute force attacks due to the lack of security verification on the login page into the system and the attack experiment shows that the attack rate for the default user level shows 100% success and for users who have changed their passwords shows 30% success so that the use of complex passwords will be safer against brute force attacks than using default passwords.

There are several limitations to this study, for example: One) The success of a dictionary brute force attack is highly dependent on the quality and completeness of the wordlist. If the wordlist does not cover the variations of passwords used in the target system, the attack may fail to detect actual vulnerabilities. Two) Brute force, especially in exhaustive mode (without a wordlist), takes a very long time for complex passwords. The study may be limited to a short test time, so it does not cover all possible combinations. Three) Researchers suggest different research methods in conducting research such as the ISSAF (Information System Security Assessment) method to obtain more accurate results.

This research is expected to provide several implications, namely: One) the discovery of vulnerabilities in the authentication mechanism indicates the need for improvements to password policies, such as implementing minimum lengths (12+ characters), character complexity, and periodic updates to prevent dictionary attacks. Two) The IT team on intelligent systems needs to perform additional verification of users who want to log into the system to minimize or prevent attempted attacks carried out by the system or robots.

Acknowledgments

-

References

1. Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, 65-88.
2. Alkhwaja, I., Albugami, M., Alkhwaja, A., Alghamdi, M., Abahussain, H., Alfawaz, F., ... & Min-Allah, N. (2023). Password cracking with brute force algorithm and dictionary attack using parallel programming. Applied Sciences, 13(10), 5979.
3. Alkhudhayr, F., Alfarraj, S., Aljameeli, B., & Elkhdiri, S. (2019, May). Information security: A review of information security issues and techniques. In 2019 2nd international conference on computer applications & information security (ICCAIS) (pp. 1-6). IEEE.
4. Al Sharaa, B., & Thuneibat, S. (2024). Ethical hacking: real evaluation model of brute force attacks in password cracking. Indonesian Journal of Electrical Engineering and Computer Science, 33(3), 1653-1659.
5. Ayankoya, F., & Ohwo, B. (2019). Brute-force attack prevention in cloud computing using one-time password and cryptographic hash function. International Journal of Computer Science and Information Security (IJCSIS), 17(2), 7-19.
6. Blakley, B., McDermott, E., & Geer, D. (2001, September). Information security is information risk management. In Proceedings of the 2001 workshop on New security paradigms (pp. 97-104).
7. Del-Valle-Soto, C., Briseño, R. A., Valdivia, L. J., & Nolazco-Flores, J. A. (2024). Unveiling wearables: exploring the global landscape of biometric applications and vital signs and behavioral impact. BioData Mining, 17(1), 15.
8. Denis, M., Zena, C., & Hayajneh, T. (2016, April). Penetration testing: Concepts, attack methods, and defense strategies. In 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-6). IEEE.
9. Edwards, D. J. (2024). Vulnerability assessment and penetration testing. In Mastering cybersecurity: Strategies, technologies, and best practices (pp. 371-412). Berkeley, CA: Apress.

10. Farahmand, F., Navathe, S. B., Sharp, G. P., & Enslow, P. H. (2005). A management perspective on risk of security threats to information systems. Information Technology and Management, 6, 203-225.
11. Florencio, D., & Herley, C. (2007, May). A large-scale study of web password habits. In Proceedings of the 16th international conference on World Wide Web (pp. 657-666).
12. Gautam, T., & Jain, A. (2015, November). Analysis of brute force attack using TG—Dataset. In 2015 SAI Intelligent Systems Conference (IntelliSys) (pp. 984-988). IEEE.
13. Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K., & Anwar, S. (2016). Intrusion response systems: Foundations, design, and challenges. Journal of Network and Computer Applications, 62, 53-74.
14. Mohammadi, V., Rahmani, A. M., Darwesh, A. M., & Sahafi, A. (2019). Trust-based recommendation systems in Internet of Things: a systematic literature review. Human-centric Computing and Information Sciences, 9, 1-61.
15. Owens, J., & Matthews, J. (2008, March). A study of passwords and methods used in brute-force SSH attacks. In USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET) (p. 8).
16. Sari, D. P., & Pakaja, F. 2024. Carrying Out Website Security Analysis Using the Standard Penetration Testing Method. International Journal of Multidisciplinary Science and Applied Research (IJOMAS). 01, 01, 22-28.
17. Shah, S., & Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. Journal of Computer Virology and Hacking Techniques, 11, 27-49.
18. Singh, H., & Sharma, H. (2020). Hands-On Web Penetration Testing with Metasploit: The subtle art of using Metasploit 5.0 for web application exploitation. Packt Publishing Ltd.
19. Sowmya, G., Jamuna, D., & Reddy, M. V. K. (2012). Blocking of brute force attack. International Journal of Engineering Research and Technology, 1(6).
20. Yaqoob, I., Hussain, S. A., Mamoon, S., Naseer, N., Akram, J., & ur Rehman, A. (2017). Penetration testing and vulnerability assessment. Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org, 7(8), 10-18.
21. Vishwakarma, R., & Jain, A. K. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommunication systems, 73(1), 3-25.