Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force

To cite this article: Toras Pangidoan Batubara *et al* 2021 *J. Phys.: Conf. Ser.* **1811** 012129

View the article online for updates and enhancements.

Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force

Toras Pangidoan Batubara[1], Syahril Efendi[2], Erna Budhiarti Nababan[3]

[1]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
[2]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
[3]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia

toraspangidoanbatubara@gmail.com

Abstract. The Bcrypt algorithm is a hashing function created from the Blowfish Algorithm by two computer security researchers, Niels Provos and David Mazieres. This hashing function has several advantages, using the original random salt (the salt is the order in which it is added to the password to make it harder to bruteforce). Random salts also prevent lookup table creation. On this basis, the authors try to do a Brute Force experiment on plaintext that has been encrypted by the Bcrypt Algorithm based on 3 characters, namely alphabetic characters, numeric characters and mixed characters to see the security results of the Bcrypt Algorithm. From the results of tests conducted, the alphabetic character with a total of 4 characters can be returned to the original plaintext within 4 days while if the number of 5 characters cannot be found the original plaintext. Then the numeric characters with a total of 7 characters can be found in the original plaintext within 10 hours. Meanwhile, for mixed characters with a total of 7 characters, the original plaintext cannot be found within 5 days. The results of this study indicate that the security performance of the Bcrypt Algorithm is very good in warding off Brute Force attacks for mixed characters while the numeric and alphabetic characters are not good enough.

1. Introduction

The Bcrypt hashing algorithm is a hashing function created from Blowfish by two security researchers, Niels Provos and David Mazières. This hashing function has several advantages, using the original random salt (the salt is the order in which it is appended to the password to make it harder to brute force). Random salts also prevent lookup table creation. Brute force attacks are among the most frequently used when it comes to collecting password data and through a dictionary attack an attacker can match as few characters as possible. Because the Bcrypt Algorithm is resistant to brute force attacks on this basis, the author tries a new method to test the security of Bcrypt in the form of a password based on character types to see the results of the security of the Bcrypt Algorithm.

2. Literatur Review

2.1. Bcrypt

Bcrypt is a password hashing with an increased number of illustrations to make it slower and lasts longer against brute-force search attacks as well as increasing computing power by combining salt to protect against rainbow table attacks. Bcrypt encrypts 192 bit hashs by using a 128 bit salt where the number of hash values is 192 bits (base-64 encoded as 31 characters) while the number of salt values is 128 bits (base-64 is encoded as 22 characters).

3. Methodology

3.1. Password security Process Using Bcrypt Algorithm

The following is an overview of the process of securing bcrypt passwords.

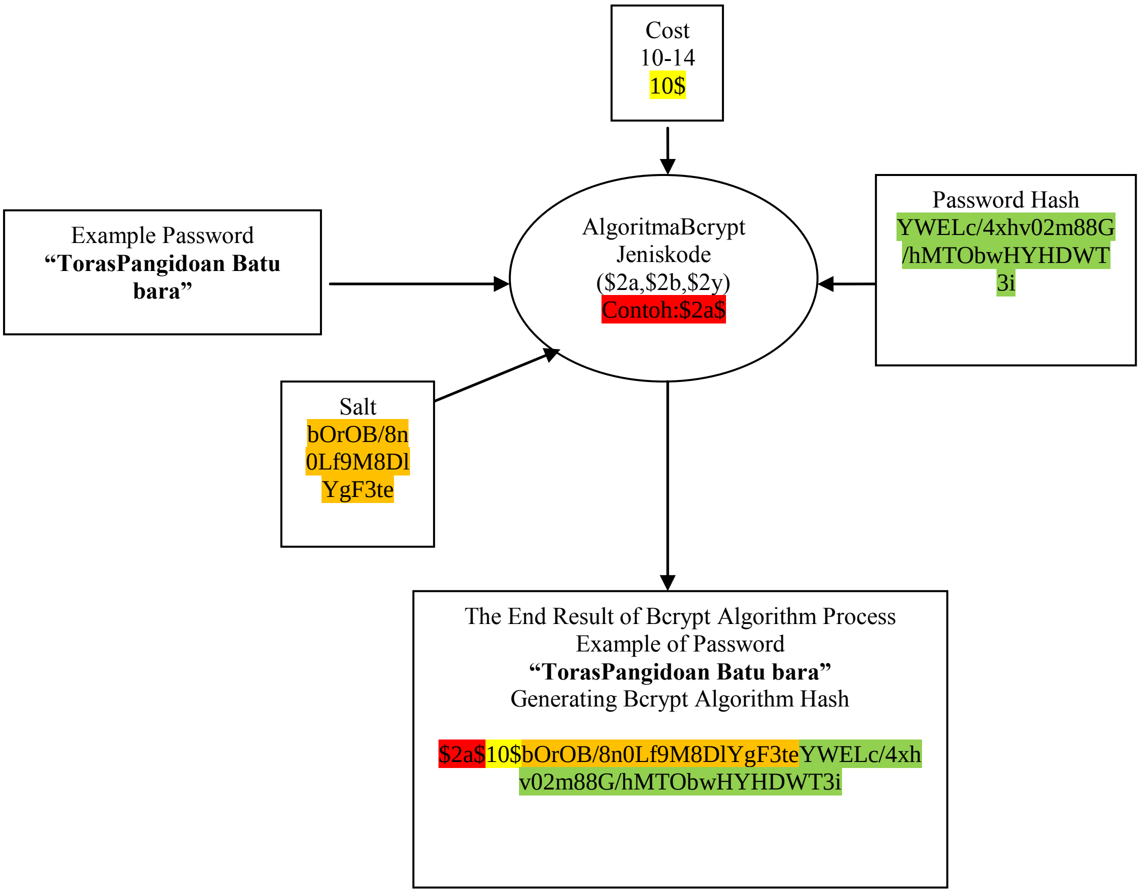
Cost
10-14
10$
Example Password
“TorasPangidoan Batu bara”
AlgoritmaBcrypt
Jeniskode
($2a,$2b,$2y)
Contoh:$2a$
Password Hash
YWELc/4xhv02m88G/hMTObwHYHDWT3i
Salt
bOrOB/8n0Lf9M8DlYgF3te
The End Result of Bcrypt Algorithm Process
Example of Password
“TorasPangidoan Batu bara”
Generating Bcrypt Algorithm Hash
$2a$10$bOrOB/8n0Lf9M8DlYgF3teYWELc/4xhv02m88G/hMTObwHYHDWT3i

Figure 1. Password security Process Using Bcrypt Algorithm

The bcrypt algorithm is:

```
Input:  (cost, salt, key)
Output: hash
state  Eks Blowfish Setup (cost, salt, key); ←
ctext    "Orphean Beholder Scry Doubt"; ←
repeat (64) begin
ctext  Encrypt ECB (state, ctext); ←
  end
  return Concatenate (cost, salt, ctext);
```

3.2. Bcrypt Brute Force Password Process Flow

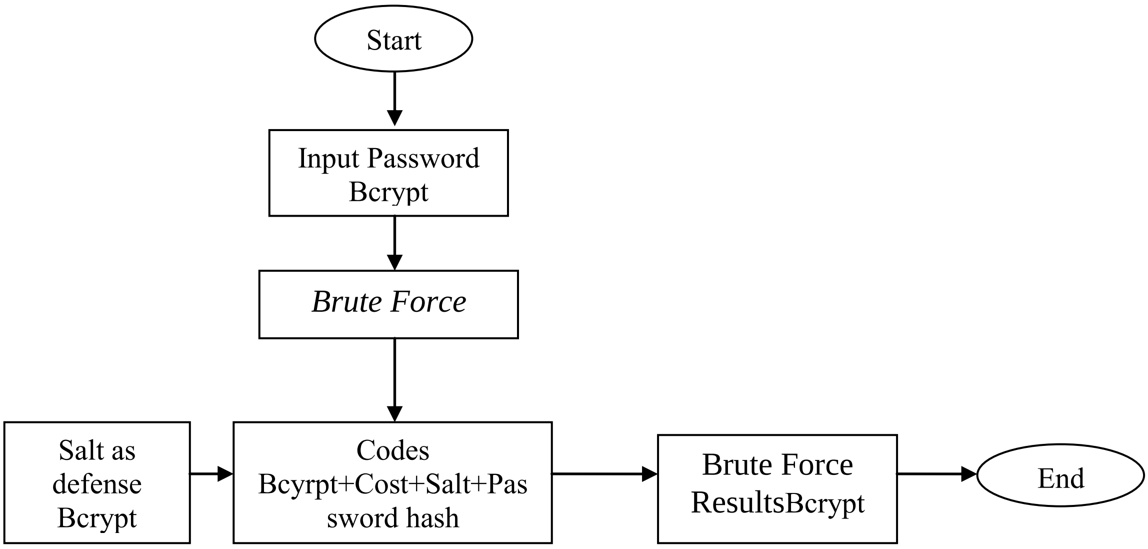

Figure 2. Bcrypt Brute Force Password Process Flow

4. Result

4.1. Brute Force Testing on Alphabetical Characters

4.1.1. Brute Force Codes $2b$ Salt 10 Alphabetical Character Passwords

Tabel 1. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2b$	Alphabet	5	10	toras	$2b$10$9eiAF6l3noJdpYrz9PQlKu2s UOU Cv/XoscmA9Pizc3Hci7UXrm4xC

Tabel 2. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2b$	Alphabet	5	$2b$10$9eiAF6l3noJdpYrz9PQlKu2sUOU Cv/XoscmA9Pizc3Hci7UXrm4xC	Not Found	-	5 days

4.1.2. Brute Force Codes $2a$ Salt 10 Alphabetical Character Passwords

Tabel 3. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2a$	Alphabet	4	10	tora	$2a$10$DEqiKH3Bha0.my4p5lWh1.2WtVk jOuz0PLD.NkJKULScGXXxs0/7G

Tabel 4. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2a$	Alphabet	4	$2a$10$DEqiKH3Bha0.my4p5lWh1.2WtVk jOuz0PLD.NkJKULScGXXxs0/7G	Found	tora	4 days

4.1.3. Brute Force Codes $2y$ Salt 10 Alphabetical Character Passwords

Tabel 5. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2y$	Alphabet	5	10	toras	$2y$10$L4rFpHl/QOubkHWWHS82De5UfP p1NosLiaBqxqyyeneuWx70WKeVO

Tabel 6. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2y$	Alphabet	5	$2y$10$L4rFpHl/QOubkHWWHS82De5UfP p1NosLiaBqxqyyeneuWx70WKeVO	Not Found	-	5 days

4.2. Brute Force Testing on Number Characters

4.2.1. Brute Force Codes $2b$ Salt 10 Numeric Character Passwords

Tabel 7. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2b$	Numbers	7	10	1234567	$2b$10$1M77axVQRHE7M9odbfIeI.esW3g/K4D6lElH9Gewh0pHqDBAyqlJG

Tabel 8. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2b$	Numbers	7	$2b$10$1M77axVQRHE7M9odbfIeI.esW3g/K4D6lElH9gEWH0pHqDBAyqlJG	Found	1234567	8 hours

4.2.2. Brute Force Codes $2a$ Salt 10 Numeric Character Passwords

Tabel 9. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2a$	Numbers	7	10	1234567	$2a$10$AUpYzEc1pmCT/OTTLFOYOe6xyqp y4f95sLGFrprl2nO/yWtWxx9Bm

Tabel 10. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2a$	Numbers	7	$2a$10$AUpYzEc1pmCT/OTTLFOYOe6xyqp y4f95sLGFrprl2nO/yWtWxx9Bm	Found	1234567	10 hours

4.2.3. Brute Force Codes $2y$ Salt 10 Numeric Character Passwords

Tabel 11. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2y$	Numbers	7	10	1234567	$2y$10$.0icA.f4fkk/c4Mu8GkO9.LuNSz Dyj.DWWepCCsBKETLoUYBFt68i

Tabel 12. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2y$	Numbers	7	$2y$10$.0icA.f4fkk/c4mU8GkO9.LuNSz dYJ.DWWepCCsBKETLoUYBFt68i	Found	1234567	10 hours

4.3. *Brute Force Testing on Mixed Characters*

4.3.1. *Brute Force $2b$ Salt 12 Mixed Character Passwords*

Tabel 13. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2b$	Mixed	7	12	toras3$	$2b$12$B3GYY8jB/.kj9ealyuhaye20inr VCDaC1vPkkcH1s95bgnyV3oluO

Tabel 14. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2b$	Mixed	7	$2b$12$B3GYY8jB/.kj9ealyuhaye20inr VCDaC1vPkkcH1s95bgnyV3oluO	Not Found	-	5 days

4.3.2. *Brute Force $2a$ Salt 10 Mixed Character Passwords*

Tabel 15. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plaintext Password	Password Results
$2a$	Mixed	7	10	toras2$	$2a$10$e9QzH0VTWujn.38u9Zm0hOv8/Ij SnLOquij1d8iQqVLzBFJPWUkj2

Tabel 16. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2a$	Mixed	7	$2a$10$e9QzH0VTWujn.38u9zM0hOv8/iJ SnLOquij1d8iQqVLzBFJPWUkj2	Not Found	-	5 days

4.3.3. Brute Force $2y$ Salt 10 Mixed Character Passwords

Tabel 17. Encryption Results

Algorithm Code	Character Type	Number Of Characters	Cost	Plantext Password	Password Results
$2y$	Mixed	7	10	toras2#	$2y$10$HLE2rlPU82rHOrASasUm/ez/Jj.jWrb70GvKRkZ1sNwwAOX6lFeJy

Tabel 18. Brute Force Results

Algorithm Code	Character Type	Number Of Characters	Password Bcrypt	Brute Force Results	Password	Time
$2y$	Mixed	7	$2y$10$HLE2rlPU82rHOrASasUm/ez/Jj.jWrb70GvKRkZ1sNwwAOX6lFeJy	Not Found	-	5 days

5. Conclusion

Based on the results of the research obtained, that the results of the performance of the Bcrypt Algorithm are quite good in warding off Brute Force attacks on alphabetic and mixed characters while for Bcrypt numeric characters it is not good enough in preventing brute force attacks.

References

[1] Provos, N., &Mazieres, D. (1999). Bcrypt algorithm. In *USENIX*
[2] Kamal, P. (2019). Security of Password Hashing in Cloud. *Journal of Information Security*, *10*(02), 45.
[3] Online Hash Crack, Professional Password Recovery.(Online)https://onlinehashcrack.com/ (25 Juni 2020)
[4] Bcrypt Generator.(Online) https://bcrypt-generator.com/ (25 Juni 2020)
[5] DailyCred, Bcrypt Calculator.(Online) https://www.dailycred.com/article/bcrypt-calculator (25 Juni 2020)
[6] P. Sriramya, &R. A. Karthika,"Providing password security by salted password hashing using bcrypt algorithm", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 13, pp. 5551-5556, 2015.
[7] Ertaul, L., Kaur, M. &Gudise, V. 2016. Implementation and Performance Analysis of PBKDF2, Bcrypt, Scrypt Algorithms. *Proceedings of the International Conference on Wireless Networks (ICWN)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016. p. 66.
[8] Kumar, N. & Chaudhary, P. 2018. Password Security Using Bcrypt with AES Encryption Algorithm. *Smart Computing and Informatics*. Springer, Singapore, 2018. p. 385-392.
[9] Bošnjak, L., Sreš, J., &Brumen, B. 2018. Brute-force and dictionary attack on hashed real-world passwords. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1161-1166). IEEE.
[10] Quyuh H Dang. Secure Hash Standard. Technical Report, 2015.