Algorithm To Ensure And Enforce Brute-Force Attack-Resilient Password In Routers

Mohammed Farik, ABM Shawkat Ali

Abstract: Issues of weak login passwords arising from default passwords in wired and wireless routers has been a concern for more than a decade. In this research we develop and test an algorithm to ensure and enforce passwords in routers that are resistant to brute-force attack. A comparative analysis is performed to show the improved strengths of passwords derived via this algorithm. Implementation of this algorithm in routers will ensure setup of brute-force attack resistant passwords.

Index Terms: Algorithm, brute-force, entropy, passwords, 802.11ac router

1 Introduction

PASSWORD authentication is a common feature in all routers for access control and administration of the router, and subsequently of the network. Lately, with an influx of 802.11ac *small office home office* (SOHO) routers that have communications compatibility with mobile devices in our society, security concerns are all-time high. While WPA2 encryptions can be enabled, if password is not strong enough, hackers will still be able to break the password, if not with easier methods, then with a time consuming complex and computation-intensive brute-force attack technique or tool. So, the idea is to create a password that is hard to break for a hacker, but not difficult to remember for the router administrator. The aim is to develop an algorithm that allows setup of only brute-force resistant password into the router.

2 Literature Review

Password issues in routers have been a matter of concern over the many years of its existence [1], [2], [3], [4]. Three parameters – length, cardinality, and entropy decide the resilience of a password against brute-force attack [5]. The current minimum password length requirement for wireless security is eight characters (Fig.1), while the default password for router login can even be zero character (no password). These need revision to 12 characters [6], [7].

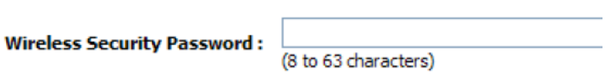

Fig.1. *Password Setup in 802.11ac Router [7]*

Furthermore, none of the default passwords in current routers have an ideal highest cardinality score of 94 [7], [8]. A score of 94 means the password is chosen from a pool of 94 characters comprising of uppercase and lowercase alphabets, numbers, and special characters Moreover, entropy is a password's measure of strength in bits, and is calculated as Equation (1) [9]. It helps estimate the number of guesses needed to break a password. Using equation (1), a password of 8 character-lengths and cardinality of 94 will be equal to 52.4 bits entropy.

$$Entropy = \log_2 N^L \qquad (1)$$

Furthermore, Equation (2) in Fig.2 represents a linear regression equation of entropies of over one-thousand default passwords [7], [8]. Even though the graph shows a few password instances having greater than 52.4 bit entropy, they cannot be regarded as strong because of the fact that they are unnecessarily lengthy and have less than 94 cardinality score. Lengthy passwords are difficult to remember without a passphrase, and usage of passphrase for example a popular song, can reveal a pattern to the hacker using easier and other decoding and cracking methods than brute-force attack technique.

$$y = 0.0469x - 2.0834 \qquad (2)$$

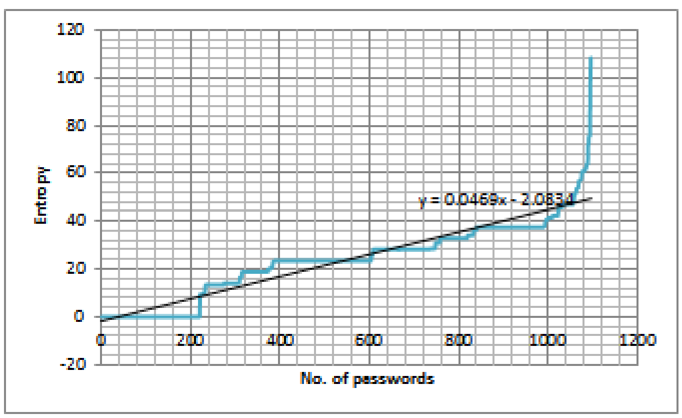

Fig.2. *Linear Regression graph on Entropies of Router Default Passwords [7]*

Fig.3 shows the time-frames within which default passwords of routers in Fig.2 can be successfully cracked using brute-force attacks.

- *Mohammed Farik is a Lecturer in Information Technology in the School of Science and Technology at The University of Fiji, PH-679-6640600. E-mail: mohammedf@unifiji.ac.fj*
- *ABM Shawkat Ali is a Professor in Information Technology in the School of Science and Technology at The University of Fiji.*

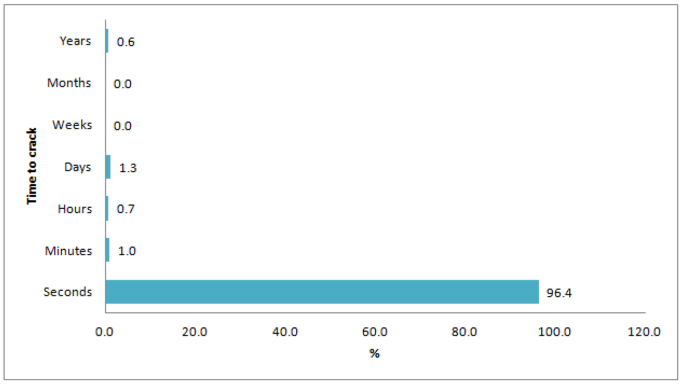

***Fig.3.** Distribution of No. of Passwords with Time Needed to Crack [7]*

It has been proven that minimum 8-character password with the cardinality score of 94 equaling an entropy value of 52.4 bits is insecure and needs changing to at least 12 characters [7]. PasswordStrengthCalculator.org (Fig.4) can be used by administrators to test their passwords for survivability against brute-force attacks [10].

***Fig.4** PasswordStrengthCalculator.org*

While an 8-character, 94 cardinality, and 52.4-bit entropy password will be cracked in 0.07 seconds by a supercomputer and in 20 minutes by a PC & GPU (Farik & Ali, Analysis of Default Passwords in Routers against Brute-Force Attack, 2015), a 12-character, 94 cardinality, and 78.7-bit entropy password will take a supercomputer 55 days, and a PC & GPU 3018 years to crack (Fig.4). In comparison, an 11 character password of *cardinality=94* can be cracked within 14 hours. Hence, not less than a minimum of 12-character and 94 cardinality should be applied in new passwords wherever needs to be set.

3 METHODOLOGY

Firstly, a password meter algorithm is developed to be implemented in a router using C++ programming language. This password meter will have the enhanced functionality to enforce entry of only strong passwords in a router. Secondly, the new router password meter application is tested for performance during password entry. Next, pre-test and post-test passwords are collected and compared. Statistical analysis shows how significant an improvement the new strong password enforcing application has brought about and whether the application is worthy of being incorporated in routers for router security improvement by means of strong passwords.

4 ALGORITHM

Software design process coverts requirements into an algorithm that can be coded. *So how does this algorithm decide if password is weak or strong?* In this algorithm, for a password to be declared a brute force resistant (strong password) and be accepted by the router, the password should pass all three conditions of *length>=12*, *cardinality>=94*, and *Entropy>=78.6* (Fig. 5). There are two paths leading from each condition – *Yes* and *No*. A failure (*No*) in any one condition will lead to declaration of a weak password, and non-acceptance by router.

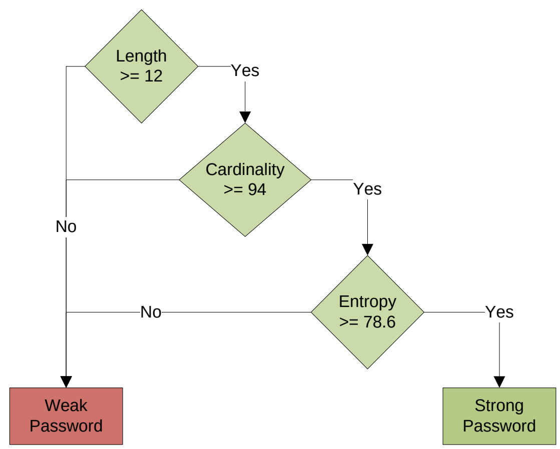

***Fig. 5** Decision Tree of algorithm's logic*

A sample implementation of the above logic is demonstrated in the proposed solution – a password meter and strength enforcer algorithm (Fig.6). The algorithm allows for the program to repeat for another entry of password, if it discovers the user is entering a weak password (Line 1 and line 13). Before, the program repeats, it displays 3 lines of message (Lines 10- 12). Line 10 informs the user that password is weak. Line 11 displays to the user the cardinality, length, and entropy of the password being entered in relation to the minimum requirements that must be met. Line 12 tells the user that the password has not been accepted, and executes line 13 to go for another session beginning with line 1.

```
1  While (Length <12 AND Cardinality <94 AND Entropy <78.6)
2    Input password of 12-63 characters
3      do
4        Count uc, lc, numbers, special.ch
5        Count length
6        Calculate cardinality (uc + lc + numbers + special.ch)
7      while (not end of string)
8    Calculate Entropy (E=log2N^L)
9    if ((Entropy<78.6) OR (Cardinality<94) OR (Length<12)) THEN
10     Display "Weak Password"
11     Display Length (min:12), Cardinality(min:94), Entropy(min:78.6 bits)
12     Display "Password Not Accepted"
13 Repeat
14   if ((Entropy>=78.6) AND (Cardinality>=94) AND (Length>=12)) THEN
15     Display "Strong Password"
16     Display Length (min:12), Cardinality(min:94), Entropy(min:78.6 bits)
17     Display "Password Accepted"
18 Stop
```

Fig. 6 *Sample Password Meter & Strength Enforcer Algorithm*

The program does not repeat if the password being entered is strong. In this case, the program runs lines 15, 16 and 17. The program prints again three messages. The first message (line 15) is "strong password". The second message is again, the one that informs the user on the cardinality, length, and entropy of the password being entered in relation to the minimum cardinality, length, and that must be met (line 16). And the last message is from line 17 that informs the user that password has been accepted. After, this the program ends (line 18). While the user is entering the password (line 2), another small while loop (lines 3 - 7) collects the characters, and analyses it for cardinality scores in lines 4 and 6, and count length of password in line 5. Line 6 calculates the cardinality by adding all cardinality counts collected in line 4. Line 8 calculates password entropy. A high cutoff value of 78.6 bits has been set for acceptable password. This value is also possible if *length>12* and *cardinality<94*. Therefore for an ideal password, one that can be declared strong and be accepted by the router, all three criteria should be fulfilled. That is, the entropy should be greater than or equal to 78.6, cardinality should be greater than or equal to 94, and length of password should be greater than or equal to 12 (lines 1 and 9). If any of the three conditions fails (lines 1 and 9), the password is not accepted and program repeats. For example, in Table 1, a password of length 13 and entropy 84.8 (even though more than 78.6) is rejected because the cardinality is 92 (less than 94).

5 TESTING

The testing process focuses on correcting errors, both in the syntax and logic of the software, and to ensure that during password entry, router will accept or reject the password based on the algorithm logics in Fig. 5 and Fig. 6, and as indicated in Table 1.

TABLE 1

FEATURES OF SAMPLE PASSWORDS TO TEST NEW ALGORITHM

Length	Cardinality	Entropy	Class
8	94	52.4	Weak/Rejected
9	94	59	Weak/Rejected
10	94	65.5	Weak/Rejected
11	94	72.1	Weak/Rejected
11	92	71.8	Weak/Rejected
12	92	78.3	Weak/Rejected
13	92	84.8	Weak/Rejected
14	92	91.3	Weak/Rejected
15	92	97.9	Weak/Rejected
12	94	78.7	Strong/Accepted
13	94	85.2	Strong/Accepted
14	94	91.8	Strong/Accepted
15	94	98.3	Strong/Accepted
16	94	104.9	Strong/Accepted
17	94	111.4	Strong/Accepted
18	94	118	Strong/Accepted
19	94	124.5	Strong/Accepted
20	94	131.1	Strong/Accepted
21	94	137.6	Strong/Accepted
22	94	144.2	Strong/Accepted
23	94	150.8	Strong/Accepted
24	94	157.3	Strong/Accepted
25	94	163.9	Strong/Accepted
26	94	170.4	Strong/Accepted
27	94	177	Strong/Accepted
28	94	183.5	Strong/Accepted
29	94	190.1	Strong/Accepted
30	94	196.6	Strong/Accepted
31	94	203.2	Strong/Accepted
32	94	209.7	Strong/Accepted

Fig. 7 *shows a scenario when a weak password is entered.* When a weak password is entered the program prints a message *"Weak password*" together with entered and expected length, cardinality, and entropy information. Moreover, it tells the user "*Password Not Accepted*" and to "*Please Try Again*". After this it displays another message "*Press any key to continue*…". Pressing any keyboard key repeats the program for a fresh entry of password.

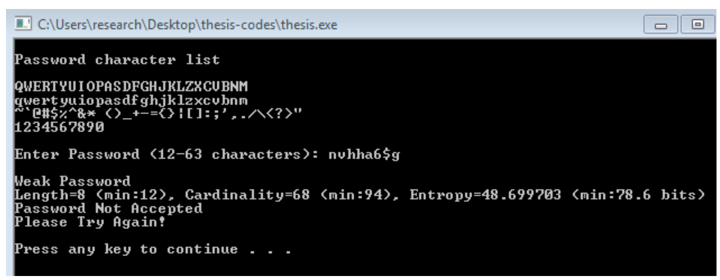

Fig. 7 *Password Rejected (Weak password)*

Likewise, when a strong password is entered (Fig.8), the program prints a message "Strong password" together with entered and expected length, cardinality, and entropy information. Moreover, it tells the user "Password Accepted". After this it displays another message "Press any key to continue…". Pressing any key ends execution of this program.

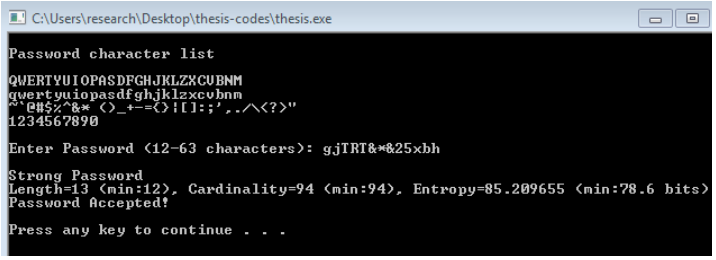

Fig. 8 *Password Accepted (Strong password)*

Moreover, when tested for classification accuracy using *J48 (C4.5)* classifier on training set data (Table 1) in *Weka*, Fig. 9 reveals 100% accuracy in regards to *correctly classified instances.*

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          30              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               30

=== Confusion Matrix ===

  a  b   <-- classified as
  9  0 |  a = Week/Rejected
  0 21 |  b = Strong/Accepted
```

Fig. 9 *J48 Classifier Output on Table 1 data*

6 Analysis

In this section, the 21 distinct entropies of possible passwords that are *accepted* via new algorithm (see Table 1) are statistically analyzed in comparison with 56 distinct entropies in router's default passwords.

Fig. 10 shows the maximum, minimum, mean, standard deviation and histogram of 56 distinct entropies of default passwords.

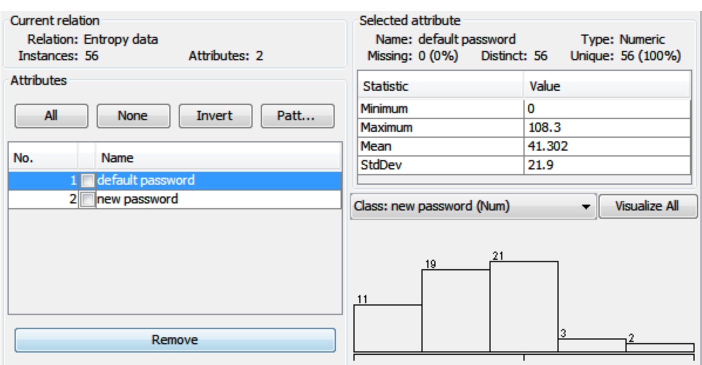

Fig. 10 *Statistics on 56 Distinct Default Password Entropies*

Fig. 11 shows the maximum, minimum, mean, standard deviation and histogram of first 21 distinct entropies of default passwords.

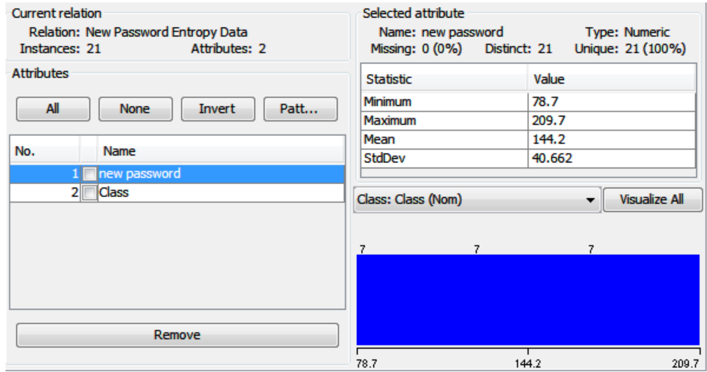

Fig. 11 *Statistics on 21 Distinct Password Entropies from New Algorithm*

It can be seen from Fig. 10 and Fig. 11 that there is significant improvement in all aspects of statistics – the minimum, maximum, mean, standard deviation and the distribution on histogram in the use of the new algorithm. Moreover, the *Class* attribute in Fig. 12 shows that only one class of passwords can be *accepted* via the new algorithm, and that is *strong*.

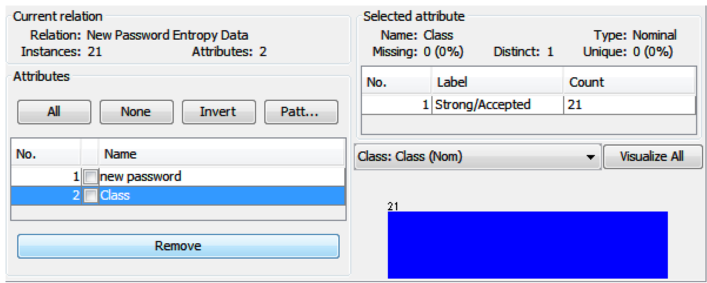

Fig. 12 *Class of Password accepted via New Algorithm*

Lastly, the line graph (Fig.13) visualizes the vast improvement gained in entropy upon utilizing the new password algorithm in composing the passwords. Distinct entropy of *default passwords* are represented by the linear Equation (3) while distinct entropy from *new algorithm passwords* are represented by the linear Equation (4).

$$y = 1.292x + 4.48 \tag{3}$$

$$y = 6.5532x + 72.114 \tag{4}$$

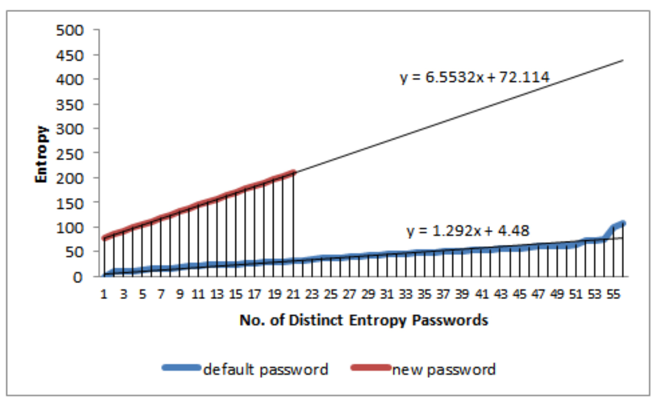

Fig. 13 *Entropy comparison of default and new algorithm passwords*

7 CONCLUSION

It can be positively concluded that the new algorithms ensure and enforce brute-force attack-resilient passwords. The algorithm prevents entry of passwords that fail even a single condition. For a password to be accepted, it should pass all the three conditions of length, cardinality, and entropy. Classification tests carried out on the algorithm using sample test data reveal 100% classification accuracy in accepting a strong password, and in rejecting a weak password. Analysis has shown significant improvements in passwords accepted via the new algorithms when compared to current default router passwords. Moreover, 100% of the passwords that were accepted via this algorithm proved to be strong and unbreakable using brute-force method on a GPU based PC. Hence, if this program is built-in into routers, the passwords that will be entered cannot be broken using brute-force methods using a GPU-based PC. It also means that the problem of weak passwords, default or otherwise that were prevalent since the beginning of routers in both wired and wireless can now be non-existent.

8 FUTURE WORK

This idea can be in the future built-in into routers by integrating it with router configuration management software. Also, the algorithm can be modified for added functionality such as to include a dashboard for the user to see the strength of entered password in real-time visualization parameters such as expected minimum time to break password. This idea can also be integrated into administrative policies by network administrators in the issuance of login passwords for *Windows Network Clients, Moodle, Email, social networks,* and other such accounts where passwords are applied.

REFERENCES

[1] D. Florencio and C. Herley, “A large-Scale Study of Web Password Habits”, Microsoft Research, Proc. WWW 2007, Banff, BC. [Online]. Available: http://research.microsoft.com/pubs/74164/www2007.pdf [Accessed 10 April 2014].

[2] M. Choi, RJ. Robles, C. Hong), T. Kim, “Wireless Network Security: Vulnerabilities, Threats, Countermeasures”, International Journal of Multimedia and Ubiquitous Engineering, Vol. 3, No. 3, July, 2008.

[3] E.N. Lorente, C. Meijer, R. Verdult, “Scrutinizing WPA2 Password Generating Algorithms in Wireless Routers”. [Online]. Available: https://www.usenix.org/system/files/conference/woot15/woot15-paper-lorente.pdf [Accessed 10 April 2014].

[4] P. Szewczyk, “The ADSL Router Forensics Process”, Edith Cowan University Research Online, ECU Publications Pre. 2011, 2010. [Online]. Available: http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=7481&context=ecuworks [Accessed 10 April 2014].

[5] M. Farik, "Improving Network Security: An Alogrithm to Enforce Strong Router Password," a minor thesis accepted in partial fulfilment of MInfTech degree at The University of Fiji, 2014.

[6] M. Farik and A. Ali, "Recurrent Security Gaps in IEEE 802.11ac Routers," International Journal of Scientific and Technology Research, vol. 4, no. 9, 2015.

[7] M. Farik and S. Ali, "Analysis of Default Passwords in Routers against Brute-Force Attack," International Journal of Scientific and Technology Research, vol. 4, no. 9, 2015.

[8] "RouterPasswords," [Online]. Available: http://www.routerpasswords.com. [Accessed 10 April 2014].

[9] "Interpreting the Calculation," [Online]. Available: http://passwordstrengthcalculator.org/interpret.php. [Accessed 10 April 2014].

[10] "Estimate password strength and survivability," [Online]. Available: http://passwordstrengthcalculator.org/index.php. [Accessed 10 April 2014].