IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

Cyber Security through Password Management Strategies

Monika Varshney[1*], Azad Kumar Shrivastava[2], Alok Aggarwal[3], Adarsh Kumar[3]

[1] of Computer Science, Mewar University, Chittorgarh (Raj), India
[2] Department of Computer Science, Mewar University, Chittorgarh (Raj), India
[3]School of Computer Science, University of Petroleum & Energy Studies, Dehradun, India

[*]*Corresponding Author:* monikafpc@gmail.com

Available online at: www.ijcseonline.org

Accepted: 26/Dec/2018, Published: 31/Dec/2018

***Abstract* -** Out of various cyber security measures password is one the most crucial measure especially due to exponential growth of Internet and multi-media users after the advent of these services over mobile gadget. This work focusses on password management strategies in context of cyber safety. Different password cracking techniques have been used by hackers in past which had been varying with the changing paradigms related to population growth and its literacy level. Recent password cracking techniques used in 2017 are analyzed. Permutation and combinations used for password strength are discussed with a mathematical model which gives how long will it take to crack a password over different machines. Two case studies for password operations have been discussed by taking Windows and Unix operating systems. For making a password a strong unbreakable password hashing & salting techniques are analyzed.

***Keywords* -** Cyber security, password management, password cracking, hashing, salting

I. INTRODUCTION

Cyber-security is a common term used to describe a set of practices, measures and/or actions to protect personal information and the computer from attacks. Major cyber security actions are install OS/software updates, run anti-virus software, prevent identity theft, turn on personal firewalls, avoid spyware/adware, protect passwords, back up important files [1]-[2]. This paper deals with password security only. Password protection has been a very essential task particularly since the emergence of smart phone over which Internet and multi-media data have been transacted. Computer since its emergence, last 70 years, to the form what we see today could penetrate only 10% of world population. Mobile phones had penetrated more than the world population only in 20 years. World population is coming to Internet and multi-media services through mobile phones exponentially, for financial transactions too. Cyber security through password has become quite vital and challenging too. Let us see how often do Facebook accounts get hacked. About 1 billion people log onto Facebook each day and adding roughly another 1 billion logons each day from twitter, instagram, linkedIn, g+ etc. [3]. Altogether, about 2 billion logins to most popular social network are taking place daily. Facebook accounts are hacked 600,000 times daily during users' login, the social working site conceded this week. The Internet powerhouse said that it records more than 1 billion logon each day, and that .06% of those logon are compromised [4].

This work focusses on password management strategies in context of cyber security. Rest of the paper is organized as follows. Recent password cracking techniques used in 2017 are analyzed in section 2. Permutation and combinations used for password strength are discussed with a mathematical model which gives how long will it take to crack a password over different machines in sections 3 & 4. Section 5 gives the two case studies for password operations by taking Windows and Unix operating systems. Section 6 analyses different security levels that a user can go through for having an unbreakable password. Hashing & salting techniques are analyzed in section 7. Section 8 concludes the work by giving few recommendations for maintaining a strong password for individuals and for individuals responsible for the design and implementation of systems.

II. THE TOP TEN PASSWORD-CRAKING TECHNIQUES USED BY HACKERS IN 2017

The top ten password-cracking techniques used by hackers in 2017 have been dictionary attack, brute force attack, rainbow table attack, phishing, social engineering, malware, offline cracking, shoulder surfing, spidering and guess.

Dictionary attack

The dictionary attack uses a simple file containing words that can be found in a dictionary, hence its rather straightforward name. In other words, this attack uses exactly the kind of words that many people use as their password. Cleverly grouping words together such as "letmein" or "superadministratorguy" will not prevent

password from being cracked this way – well, not for more than a few extra seconds.

Brute force attack

Similar to the dictionary attack, the brute force attack comes with an added bonus for the hacker. Instead of simply using words, a brute force attack lets them detect non-dictionary words by working through all possible alpha-numeric combinations from aaa1 to zzz10. It's not quick, provided the password is over a handful of characters long, but it will uncover password eventually. Brute force attacks can be shortened by throwing additional computing horsepower, in terms of both processing power – including harnessing the power of the video card GPU – and machine numbers, such as using distributed computing models like online bitcoin miners.

Rainbow attack

Most modern systems now store passwords in a hash. This means that even if someone can get to the area or file that stores the password, what they get is an encrypted password. One approach to cracking this encryption is to take dictionary file and hash each word and compare it to the hashed password. This is very time and CPU intensive. A faster approach is to take a table with all the words in the dictionary already hashed and compare the hash from the password file to the list of hashes. If there is a match, hacker now know the password.

Phishing

There's an easy way to hack: ask the user for his or her password. A phishing email leads the unsuspecting reader to a faked login page associated with whatever service it is the hacker wants to access, requesting the user to put right some terrible problem with their security. That page then skims their password and the hacker can go use it for their own purpose.

Social engineering

Social engineering takes the whole "ask the user" concept outside of the inbox that phishing tends to stick with and into the real world. A favorite of the social engineer is to call an office posing as an IT security tech guy and simply ask for the network access password. It could be amazed at how often this works. Some even have the necessary gonads to don a suit and name badge before walking into a business to ask the receptionist the same question face to face.

Malware

A keylogger, or screen scraper, can be installed by malware which records everything we type or takes screenshots during a login process, and then forwards a copy of this file to hacker central. Some malware will look for the existence of a web browser client password file and copy this which, unless properly encrypted, will contain easily accessible saved passwords from the user's browsing history.

Offline Cracking

It's easy to imagine that passwords are safe when the systems they protect lock out users after three or four wrong guesses, blocking automated guessing applications. Well, that would be true if it were not for the fact that most password hacking takes place offline, using a set of hashes in a password file that has been 'obtained' from a compromised system.

Shoulder surfing

The most confident of hackers will take the guise of a parcel courier, aircon service technician or anything else that gets them access to an office building. Once they are in, the service personnel "uniform" provides a kind of free pass to wander around unhindered, and make note of passwords being entered by genuine members of staff. It also provides an excellent opportunity to eyeball all those post-it notes stuck to the front of LCD screens with logins scribbled upon them.

Spidering

Savvy hackers have realised that many corporate passwords are made up of words that are connected to the business itself. Studying corporate literature, website sales material and even the websites of competitors and listed customers can provide the ammunition to build a custom word list to use in a brute force attack.

Guess

The password crackers best friend, of course, is the predictability of the user. Unless a truly random password has been created using software dedicated to the task, a user-generated 'random' password is unlikely to be anything of the sort.

III. PERMUTATIONS & COMBINATIONS OF PASSWORD

To calculate combinations, we will use the formula

$^{n}C_{r} = n! / r! * (n - r)!$

where, n is the total number of items, r is the number of items being chosen at a time. Table 1 shows possible combinations for different character sets.

Table 1: Possible combinations for different character sets

Character Sets used in Password	Calculation	Possible Combinations

Lowercase Alpha Set only	26^8	208,827,064,576
Full Alpha Set	52^8	53,459,728,531,456
Full Alpha + Number Set	62^8	218,340,105,584,896
Full Set of allowed printable characters set	(10+26+26+19)^8	645,753,531,245,761

IV. HOW LONG WOULD IT TAKE TO CRACK THE PASSWORD

(i) Includes letters and numbers, no upper- or lower-case and no symbols

For 6 characters 2.25 billion possible combinations.

- Cracking online using web app hitting a target site with one thousand guesses per second: 3.7 weeks
- Cracking offline using high-powered servers or desktops (one hundred billion guesses/second): 0.0224 seconds
- Cracking offline, using massively parallel multiprocessing clusters or grid (one hundred trillion guesses per second): 0.0000224 seconds

For 10 characters 3.76 quadrillion possible.

- Cracking online using web app hitting a target site with one thousand guesses per second: 3.7 weeks
- Cracking offline using high-powered servers or desktops (one hundred billion guesses/second): 10.45 hours
- Cracking offline, using massively parallel multiprocessing clusters or grid (one hundred trillion guesses per second): 37.61 seconds.

(ii) Add a symbol, make the crack several orders of magnitude more difficult:

For 6 characters 7.6 trillion possible combinations.

- Cracking online using web app hitting a target site with one thousand guesses per second: 2.4 centuries
- Cracking offline using high-powered servers or desktops (one hundred billion guesses/second): 1.26 minutes
- Cracking offline, using massively parallel multiprocessing clusters or grid (one hundred trillion guesses per second): 0.0756 seconds

For 10 characters, possible combinations 171.3 sextillion (171,269,557,687,901,638,419; 1.71×10^{20})

- Cracking online using web app hitting a target site with one thousand guesses per second: 54.46 million centuries
- Cracking offline using high-powered servers or desktops (one hundred billion guesses/second) 54.46 years
- Cracking offline, using massively parallel multiprocessing clusters or grid (one hundred trillion guesses per second): 2.83 weeks

V. WINDOWS PASSWORD & UNIX PASSWORD

Windows Passwords

- Set or change password → Windows generates a LM hash and a NT hash.
- Two hashing functions used to encrypt passwords
  - LAN Manager hash (LM hash)
    - Password is padded with zeros until there are 14 characters.
    - It is then converted to uppercase and split into two 7-character pieces
    - Each half is encrypted using an 8-byte DES (data encryption standard) key
    - Result is combined into a 16-byte, one way hash value
  - NT hash (NT hash)
    - Converts password to Unicode and uses MD4 hash algorithm to obtain a 16-byte value
- Hashes are stored in the Security Accounts Manager database
  - *Commonly known as " SAM" or "the SAM file"*
- *SAM is locked by system kernel when system is running.*
  - *File location: C:\WINNT\SYSTEM32\CONFIG*
- *SYSKEY*

Unix Password

- Uses modified DES as if it were a hash function
  Encrypt NULL string using password as the key (Truncates passwords to 8 characters!), Artificial slowdown: run DES 25 times, Can instruct modern UNIXes to use MD5 hash function

- Problem: passwords are not truly random
  With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are 948 ≈ 6 quadrillion possible 8-character passwords. Humans like to use dictionary words, human and pet names ≈ 1 million common passwords. On average each person has 8-12 passwords. Different systems impose different requirements on passwords. Passwords need to be changed often. Some passwords are used occasionally (once a year).

VI. DIFFERENT SECURITY LEVELS OF PASSWORD

(i) Filing System: Clear text (Highly insecure)

(ii) Dedicated Authentication Server: Clear text (Highly insecure)

(iii) Encrypted: like Password + Encryption = bf4ee8HjaQkbw (Secured upto level 1)

(iv) Hashed: like Password + Hash function = aad3b435b51404eeaad3b435b51404ee (secured upto level 2)

(v) Salted Hash: like (Username + Salt + Password) + Hash function = e3ed2cb1f5e0162199be16b12419c012 (secured upto level 3, highest security by now)

VII. HASHING & SALTING TECHNIQUES

Hashing

- Instead of user password, store hash of password
- When user enters password, compute its hash and compare with entry in password file. System does not store actual passwords!
- Hash function H must have some properties
  - One-way: given H(password), hard to find password
    - No known algorithm better than trial and error
  - Collision-resistant: given H(password1), hard to find password2 such that H(password1)=H(password2)
    - It should even be hard to find any pair p1,p2 s.t. H(p1)=H(p2)

Hashing process is shown in figure 1.

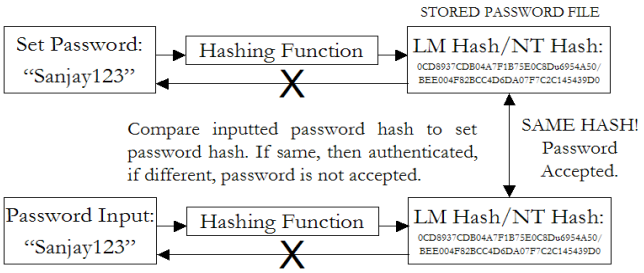

Figure 1: Hashing operation

Salting

- Salting requires adding a random piece of data and to the password before hashing it.
  - This means that the same string will hash to different values at different times
  - Users with the same password have different entries in the password file
  - Salt is stored with the data that is encrypted
- Hacker has to get the salt add it to each possible word and then rehash the data prior to comparing with the stored password.

Salting process is shown in figure 2.

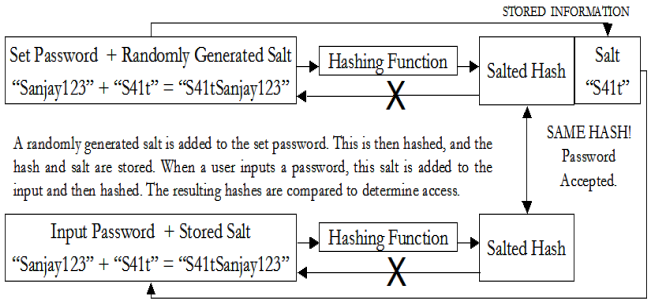

Figure 2: Salting operation

VIII. FEW RECOMMENDATIONS FOR MAINTAINING A STRONG PASSWORD

For Individuals:

- Do not share your password with anyone for any reason
- Change your password periodically
- Consider using a passphrase instead of a password
- Do not write your password down or store it in an insecure manner
- Avoid reusing a password
- Avoid using the same password for multiple accounts
- Do not use automatic logon functionality
- Don't use Dictionary words
- Use two-factor authentication

For individuals responsible for the design and implementation of systems:

- Change default account passwords
- Implement strict controls for system-level and shared service account passwords
- Do not use the same password for multiple administrator accounts
- Do not allow passwords to be transmitted in plain-text
- Do not store passwords in easily reversible form
- Implement automated notification of a password change or reset

REFERENCES

1. http://manuals.ucdavis.edu/ppm/310/310-22.htm)
2. http://security.ucdavis.edu/cybersafety.cfm
3. http://security.ucdavis.edu
4. http://security.ucdavis.edu/cybersafetybasics.cfm

Authors Profile

Monika Varshney is an Assistant Professor with Dr. Bhimrao Ambedkar University, Agra, India and enrolled in Ph.D. (C.S.E.) from , Mewar University, Gangar, Chittorgarh (Raj) India. She received her M.C.A. from IGNOU, New Delhi, India in the year 2008. Her research interest includes Data mining, Data Base Management System, Algorithm development and Decision Support System etc.

Azad Shrivastava is Professor at Department of Computer Science, Mewar University, Gangar, Chittorgarh (Raj) India. He did his Ph.D. from 'Atal Behari Vajpayee-Indian Institute of Information Technology and Management', Gwalior, Madhya Pradesh, India in the year 2009. He has an academic, research, and industry experience of about 14 years. He has been associated with CMC Ltd., TCS, AETPL. His areas of interest include Deep Learning, Machine learning, AI and NN & Big data on CPU & GPU Cluster for DWH & IOT etc.

Alok Aggarwal received his bachelors' and masters' degrees in Computer Science & Engineering in 1995 and 2001 respectively and his PhD degree in Engineering from IIT Roorkee, Roorkee, India in 2010. He has academic experience of 18 years, industry experience of 4 years and research experience of 5 years. He has contributed more than 150 research contributions in different journals and conference proceedings. Currently he is working with University of Petroleum & Energy Studies, Dehradun, India as Professor in CSE department.

Dr. Adarsh Kumar received his Master degree (M. Tech) in Software Engineering from Thapar University, Patiala, Punjab, India, in 2005 and earned his PhD degree from Jaypee Institute of Information Technology University, Noida, India in 2016 followed by Post-Doc from Software Research Institute, Athlone Institute of Technology, Ireland during 2016-2018. Currently, he is working with University of Petroleum & Energy Studies, Dehradun, India as Associate Professor in School of Computer Science.

