Scott Christensen

10/26/16

C.S. 465

Project #7: Password Cracking

1. 6-character password - I was cracking passwords that were more likely on a wordlist in about 10 seconds. However, I calculate cracking a random 6-character alpha-numeric password would take 62^6 / 18,000 = 36 days at worst case.

8-character password - I was cracking passwords that were more likely on a wordlist in just a few minutes. However, I calculate cracking a random 8-character alpha-numeric password would take 62^8 / 18,000 = 384 years at worst case.

10-character password - I calculate cracking a random 10-character alpha-numeric password would take 62^10 / 18,000 = 1,479 millennia at worst case.

12-character password - I calculate cracking a random 12-character alpha-numeric password would take 62^10 / 18,000 = 5,683,569 millennia at worst case.

2. Yes, I think that that the password meter is a very good indicator of actual password strength. I feel this way because it tracks most everything about the password from characters used to patterns in the characters themselves. It also shows how all of these factors interact together.

Based off of my results from the attacks described above, I would recommend a minimum password length of 8 characters of alpha-numeric characters, so long as you didn't use common or easily broken passwords. Even then, it could take years to crack at worst case.

3. 6-character password – Recalculating this based of the proposed power of the 4 Radeon 5970, I calculate cracking a random 6-character alpha-numeric password would take 62^6 / 33100000000 = 1.72 seconds at worst case.

8-character password - Recalculating this based of the proposed power of the 4 Radeon 5970, I calculate cracking a random 8-character alpha-numeric password would take 62^8 / 33100000000 = 109 minutes at worst case.

10-character password - Recalculating this based of the proposed power of the 4 Radeon 5970, I calculate cracking a random 10-character alpha-numeric password would take 62^10 / 33100000000 = 293 days at worst case.

12-character password - Recalculating this based of the proposed power of the 4 Radeon 5970, I calculate cracking a random 12-character alpha-numeric password would take 62^12 / 33100000000 = 3.1 millennia at worst case.

Based off of these results, I definitely think we should change the minimum password results to be at least 12 characters, because we want the average breakability to be in years, not just minutes or days to crack.

4. From what I understand, using SHA-512 is preferred over MD5 for hashing. MD5 is old, common, and there are a lot of rainbow tables against it. Plus, it is no longer secure as a cryptographic hash because you can generate different messages that hash to the same value. SHA-512 is not impervious to attacks and perhaps a tad slower. But, it is part of the SHA-2 family and is currently among the strongest when considering commonness, analysis, and security. Plus, the hash is so big, about 64 characters long to be precise, which makes it very hard to attack. So yes, I'd prefer using SHA-512 for hashing passwords.

5. Yes, using a salt increases password security. They increase the cost of offline attacks by making the attacker compute a hash from every user's salt and guessed password and it prevents duplicate password generation. Without salts, attackers can just compute the passwords and compare it against others. There is an additional step to take when comparing against users with salts.

6. No, it does not lessen the importance of offline password attack protection. Online attacks are protected because attacks can be detected and lock someone out, while there is not such protection from offline attacks. With further improvements in hardware and new methodologies developed enabling more and more hashes to be performed faster and faster. It

actually increases the importance of offline password attacks because more attacks resort to offline attacks and not online attacks.

7. Should the day come that minimal password lengths are too large for a typical person to remember, there will be far more security breaches unrelated to the algorithms employed. My thought is that should this scenario arise, more people will have to write their passwords down so there will be more security breaches based off of people having their written passwords stolen or shoulder surfing attacks since people will be spending more time typing in their passwords or some kind of social engineering attack that pretends to offer saving your long passwords to cookies so that people will not have to remember them.

My recommendation for the next step in password evolution is to increase the minimum password length to 10 or 12, mandate the use of non-alpha characters, and if possible find another secure encryption algorithm that could be employed on our passwords in addition to those currently being used. Plus, if passwords were longer, it would increase the cost of cracking those passwords by a lot.

EXTRA CREDIT

8. After downloading and comparing the cracking results given by John the Ripper in comparison to those given by oclHashCat, here are the findings. It was found that John the Ripper was a bit faster than oclHashCat when running the MD5 and the SHA-1 algorithm, which it performed at about 17,000 c/s. However, oclHashCat performed about 31% faster than John the Ripper when running the MD4 algorithm.