CSCI 530 - Security Systems Lab

Rishit Saiya - 4940905271, Lab - 2

September 23, 2022

1

a.

Without salt: The number of different ways a password can come out if hashed using no salt, is 1.

b.

Without salt: The number of entries there will be in the dictionary the cracker must create is 100,000.

c.

With salt: The number of different ways a password could come out if hashed when prefixed with a random 2-byte salt is $95 \times 95 = 9025$, assuming if the character exists on the US keyboard are 95. If all ASCII other characters are considered in bit fashion then, it is $256 \times 256 = 65,536$

d.

With salt: The number of entries will there be in the dictionary the cracker must create is $9025 \times 100,000 = 902,500,000$ assuming 95 characters on keyboard. Considering all 256 ASCII characters, then the number of different ways become $65,536 \times 100,000 = 6,553,600,000$.

e.

With salt: If all the words in the language are 8 characters long and resolve to hashes 86 bytes long, thus requiring 94 bytes (assuming 2 bytes of salt are included in here) to store each mapped pair (dictionary entry), then the number of Gigabytes (for the case of 95 US keyboard key characters) the cracker's dictionary must occupy is:

$$(94) \times \frac{902,500,000}{1024/1024/1024} \approx 79$$

This shows 79 GB (95 US keyboard characters) since we need another 2 bytes for the salt for each 94 byte entry. In case of 256 ASCII characters is considered, the number of Gigabytes (for the case of 256 ASCII characters) the cracker's dictionary must occupy is:

$$(94) \times \frac{6,553,600,000}{1024/1024/1024} \approx 573.73$$

This shows 573.73 GB (256 ASCII characters) since we need another 2 bytes for the salt for each 94 byte entry.

2

a.

The length of the numbers-only password that requires at least 50 years to crack, according to the spreadsheet, is 17 characters.

b.

Account for Moore's law. It says computing power doubles every 2 years. The spreadsheet is dated. It reflects the computing power of 10 years ago . For today, you need to increase its computing power assumptions by a factor of 32 (having doubled 5 times over the 10 years). Do so by entering 32 as the "Special factor" in cell G1 (which is applied in the "computing power" cell, E24, as a multiplier). Thus, with today's computing power, the length of the numerals-only password that requires at least the rest of your life to crack is 18 characters.

c.

Account for Moore's law's continued operation. Let's assume Moore's law doesn't stop. (There's debate about that. But let's set it aside because if Moore's law's potential to continue raising cracking power is blunted, GPU advances or specialized cracking silicon may more than fill the gap.) Then today's isn't the right computing power for the upcoming 50 years' calculations. I say that on average (less near term, more far term) the upcoming power is 2.5 million times today's (approximately). Using 2.5 million as your future computing power, the length of the password that requires at least 50 years to crack becomes 25 characters. (Multiply the current special factor by yet a further 2.5x10^6)

d.

If you then made the one change of allowing mixed random characters (spreadsheet's "PURELY Random Combo of Alpha/Numeric/Special") instead of confining your password to numerals only you should be able to use a shorter password with equal effect. The shortest "mixed character" password that'll last 50 years is 13 characters.