Gibberish v2.1

Proof of concept password generation tool

By Glenn Larsson (1973gl@gmail.com)

And yes i know that there are several other programs called “Gibberish” but i really do not care.

The idea

-"What is your password to Linkedin?"

-"I do not know."

-"What is your Facebook password?"

-"I really can not remember."

-"What is the password to your gmail account?"

-"I would love to tell you, but i can not even begin to spell it."

This is not a product, it is a proof of concept.

The idea (continued)

Gibberish generates passwords too complex to remember for normal human beings. It *guarantees* strong passwords unlike conventional password managers that just stores any password the user provides for a given site.

Gibberish generates passwords by:

- Non-sensitive entropy
- One or more simple passwords
- Entropy can be stored in the clear.

You can choose what the entropy should be,i.e. instead of typing “My secret banking password” you can enter “World of warcraft” or anything else you feel like in the input dialogue.

Similar password generators

- Apps for android. None for windows (that i found at the time).
- Apps require user to remember every piece of entropy that is used to generate passwords. Gibberish do not require that.
- Changing password in an app requires you to CHANGE the password - AND remember it.
- Gibberish combines the best of both worlds from generation to management.
- Gibberish makes it easy to change password at a click of a button, while keeping your master password/passwords.
- If Special Characters checkbox is checked, Gibberish *guarantees* the inclusion of at least two special characters. Apps that generates passwords through Base64 does not guarantee special characters at all.

Notes on Password Managers.

- Stores passwords encrypted by a password. One compromise = everything lost.
- Do not validate that passwords are strong and gives no guidance, accepts to store any crap password the user comes up with.
- Have known plaintext / structure to validate against = makes them susceptible to offline brute force attacks.
- Some autofill fields and some even click submit. This makes them vulnerable to javascript/iframe field/site impersonation attacks. ("Password Managers: Attacks and Defenses", Page 6)

Notes on Gibberish

- Gibberish generates passwords based on a entropy and a master password (or several master passwords).
- Part of the entropy is known (i.e. people KNOW you have an gmail account)
- Part of the entropy are user settings, and a generated seed.
- Entropy is not secret and is not encrypted. (It CAN be though in future versions if required = exercise for the reader.)
- Entropy is not decisive for a generated password, is merely affects the outcome.
- Entropy is injected in the hashing process, but where it ends up depends on your password for the specific site..
- Not susceptible to offline brute force attacks - there are no hashes or known plaintext to validate against. Without data from a keylogger and the entropy file, you cannot recreate the user input.
- Gibberish helps user generate strong passwords (Up to 160 bits of entropy) that are hard to crack.
- Gibberish is Open Source.
- Gibberish is free.

Algorithms used:

- Base64
- SHA256
- Separate function to generate and include certain special characters (if not user disabled)

How the hashing is done

1. Password is hashed to Pwd[]
2. Key (Byte) is created Pwd[] bytes XORed together = A value ranging from 0-255
3. Entropy is hashed to Ent[]
4. For... next (1000 times) + Key
5. Pwd[] = SHA256(Pwd)
6. If it is turn 500 + Key then // This makes entropy injection key dependant
7. Pwd[] = Pwd[] XOR Ent[]
8. When done, convert Pwd[] array to Base64 character string.
9. Inject special characters if required (User setting, checkbox) - based upon password.
10. Cut off password if the user wants a shorter one that works with a specific site, or server (40 to 160 bits = 8 to 32 bytes, see next slide for details on password strength)

Note: First 500 rounds of hashing are unaffected by entropy. Entropy is injected key dependently somewhere between round 500 to 755, this makes it a moving target and harder to crack.

A word on password strength:

The resulting strength is calculated as:

Number of bytes * 5 bits

- 8 characters = 40 bits
- 16 characters = 80 bits
- 32 characters = 160 bits

The charset A-Z + a-z + 0-9 roughly corresponds to 5 bits of entropy per generated byte.

(Note: The inclusion of special characters do increase strength even more, but this has not been taken into consideration)

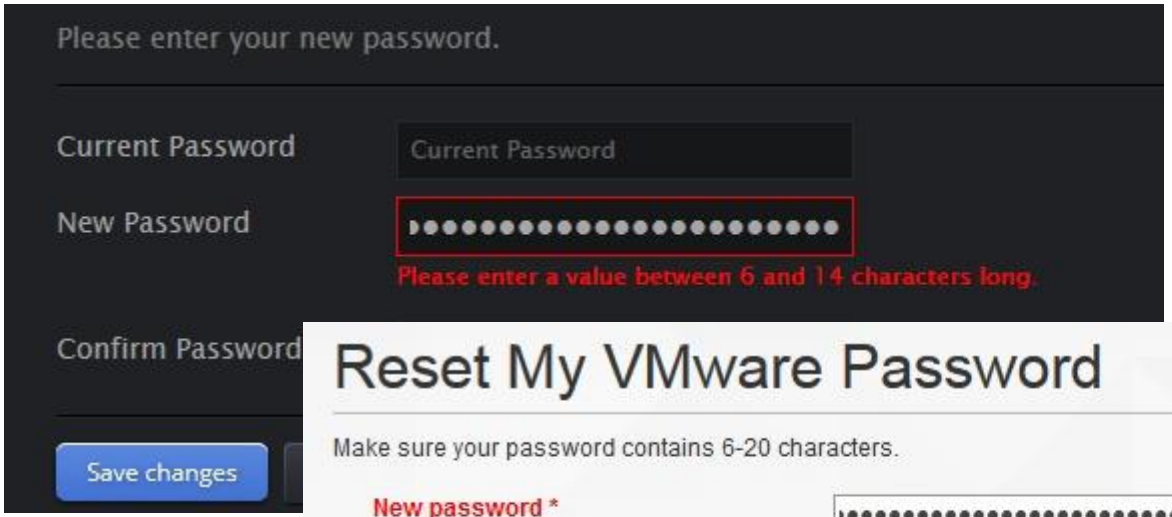
Why downgrade a password?

160 bit passwords are secure. It is not always possible to *use them*:

- Some websites have as specific field size in a password table, and cannot accept more.
- Some websites / servers are coded to only accept 8 character passwords, or a 8-16 character password.
- Some websites / servers filters special characters, probably because they are afraid of SQL injections. This is a *legitimate fear*.

In general, you should generate 160 bit (32 character) with Special Characters. This is the default strength Gibberish generates, you have to **manually decrease the security** for any given site. *(Supported sizes range from 8 to 32 characters and selectable special characters)*

Examples of limited password fields



Strengthening it further

When you start Gibberish for the first time, a unique ID is created called *LocalSeed*. This is also used as entropy. The reason why this is done is to make everyone's set of passwords more unique.

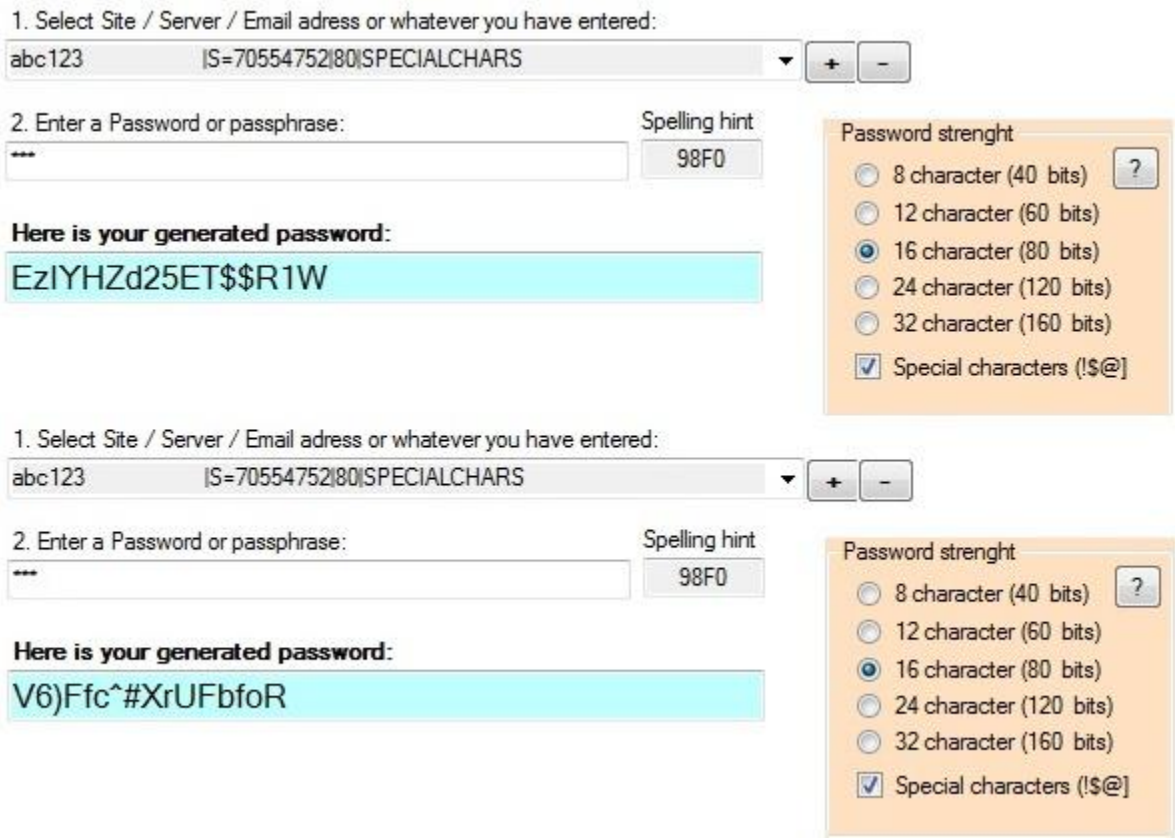
If someones master password / password combination is compromised online (i.e. a hacked site), the attacker cannot just throw a precalculated hash list at the rest of the community since every installation is unique.

LocalSeed in “action”:

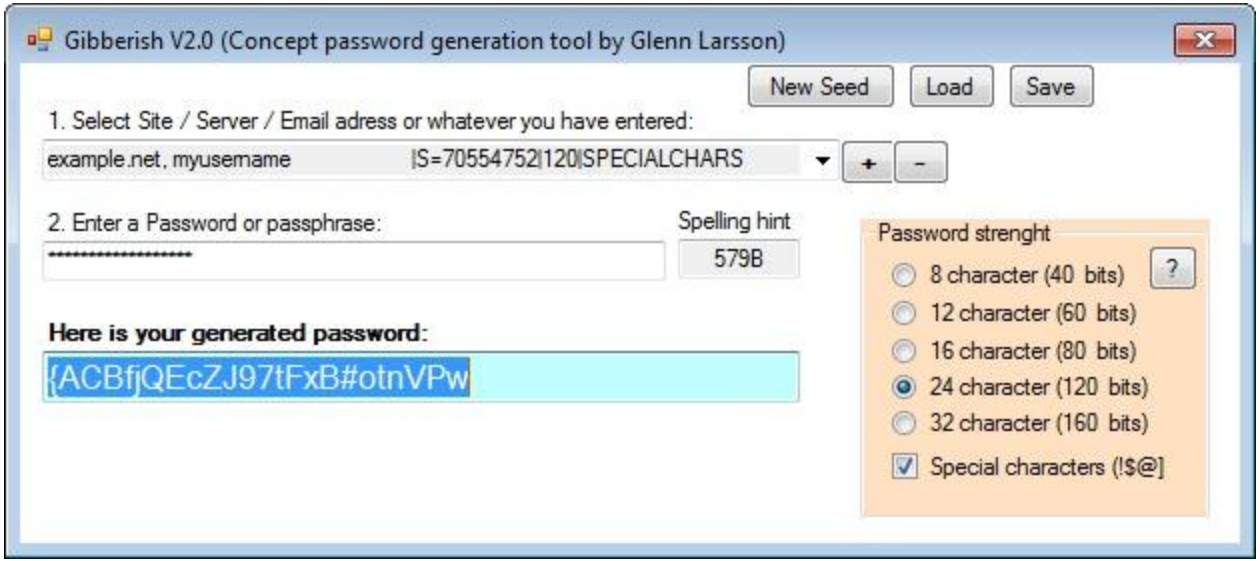
Same:

- Entropy
- Password
- Settings

Different *generated password*, due to different LocalSeed in the two instances of Gibberish.



Gibberish screen layout:



(DEMO)

Final notes:

- Gibberish remembers strength settings for each site.
- Generates a new password at the click of a button (Generate new seed). Keep your old password and generate a new one - if you like.
- A spelling hint allows you to see spelling of the password you enter without revealing it to someone watching over your shoulder.
- Password is generated on the fly as you type. Just copy / paste to wherever you want it.
- Changing password is not automatically saved, you have to save this **manually** (Save-button) **after** you changed it on the specific site. This is to prevent saving uncommitted changes following a crash or failure to update.
- If you did not want to change your password, you can (re)load it by pressing the Load-button.
- You can enter one password, unique passwords, or group passwords for specific sites, i.e. one for social media sites, one for work sites, one for online games (etc). If one password gets compromised, you do not have to change every site, unlike a password manager.
- Source code is available here: https://drive.google.com/file/d/0B6OPx7yJQjxuVzRadUl5RDhnNmM/view?usp=sharing