|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit offset | 0–3 | 4-7 | | 8–15 | | 16–31 |
| 0 | TAG | SALT | | | | |
| 32 | Destination Address | | | | | |
| 64 | Zeros | | Protocol | | TCP length | |
| 96 | Source port | | | Destination port | | |
| 128 |  | | | | | |
| 160 |  | | | | | |
| 192 |  | | | | | |
| 224 |  | | | | | |
|  |  | | | | | |

\* Analysis of Password Safe 3.x,

Password Safe [1] is a free and open source software program for storing passwords originally authored by Bruce Schneier. From a developer point of view, this format has been easiest to write code for since the database format is well documented in formatV3.txt [2]. The same database format is used by Password Gorilla as well as Pasaffe password manager. Hence the

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | File Format (Header Fields) | | | | TAG | 4 bytes | The 4 ASCII Characters ‘PWS3’ | | SALT | 32 bytes | 256 random bit value generated at file creation | | ITER | 32 bit LE value | number of rounds in the key stretch algorithm | | H(P’) | 32 bytes | SHA-256 of the user’s passphrase | | B1 | 16 bytes | Encrypted 128 random value using P’ with Twofish algorithm | | B2 | 16 bytes | Encrypted 128 random value using P’ with Twofish algorithm | | B3 | 16 bytes | Encrypted 128 random value using P’ with Twofish algorithm | | B4 | 16 bytes | Encrypted 128 random value using P’ with Twofish algorithm | | Init vector | 16 bytes | 128 bit random value that is the Initialization Vector for the content’s encryption | | Header | N bytes | General information for the database | | Records | B bytes | The records in the database |   Password Safe v3 Database Format  The Password Safe 3 format supports "variable key stretching", which is a means of protecting a database against brute-force attacks. The "iterations" (ITER) parameter indicates the complexity of the key stretching: the higher this value, the longer it takes to open a database, and the longer it takes an attacker to test one password. V2 Format Weakness In the interest of full disclosure, it should be noted that a potential weakness was discovered with the old Password Safe 2 ("V2") file format. This issue affected the "key stretching" process that is intended to slow down a brute force attack against a database's master password (i.e., repeated attempts at guessing the password). The weakness in the file format's design allowed brute force attacks 1000 times faster than intended. The number sounds worse than it is: a good, long master password is one among billions of billions of combinations, and a factor of 1000 does not make a practical difference. However, the factor may have an impact on the security of password databases that use a short, more easily guessable master password. The Password Safe 3 format avoids this issue by depending on the result of the key stretching operation (which is computationally expensive) as an input to decryption -- therefore, the operation cannot be bypassed. No similar vulnerability has been published for V3 database format. | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

References:

[1] <http://passwordsafe.sourceforge.net/>

[2] <http://passwordsafe.svn.sourceforge.net/viewvc/passwordsafe/trunk/pwsafe/pwsafe/docs/formatV3.txt?revision=4979>

<http://www.fpx.de/fp/Software/Gorilla/help.html#V3Format>