

About Google Account Password Recovery

Google account is the single centralized account used by all of the Google services such as Gmail, Picasa, GTalk, iGoogle, Desktop Search and many more. Most of the Google's desktop applications such as GTalk, Picasa etc stored the Google account password for future use. Even most of the browsers such as Internet Explorer, Firefox, Chrome store the login passwords for visited websites in their secret store. This research article throws light on the internal password storage and encryption mechanisms used for storing the Google account password by some of the prominent applications. Also it shows the methods to decrypt the Google password for each of these applications.

Password Secrets of GTalk

GTalk is the Google's instant messenger application integrated with voice and video chat feature. Like any of the Google application it uses the same Google account password and stores it for subsequent logins in an encrypted format.

It stores the account information at following location in the registry

```
HKEY_CURRENT_USER\Software\Google\Google Talk\Accounts
```

For each account, it creates separate key with the account name under this registry location as shown below.



For each stored account, the encrypted password stored in the registry value '**pw**'.

GTalk encrypts the Google account password using **Windows Cryptography functions**. Here are the step by step instructions to decrypt this password.

Enumerate through the GTalk accounts registry key and get the stored account name & encrypted password. Now get the currently logged on username & domain name of the system. Create the hash of the username and then hash the domain name on top of it to create entropy data of 16 bytes. Next hash the encrypted password with magic num

bers. Finally pass this modified password and entropy data to **CryptUnprotectData** function to decrypt the password.

Deciphering the Password from Picasa Store

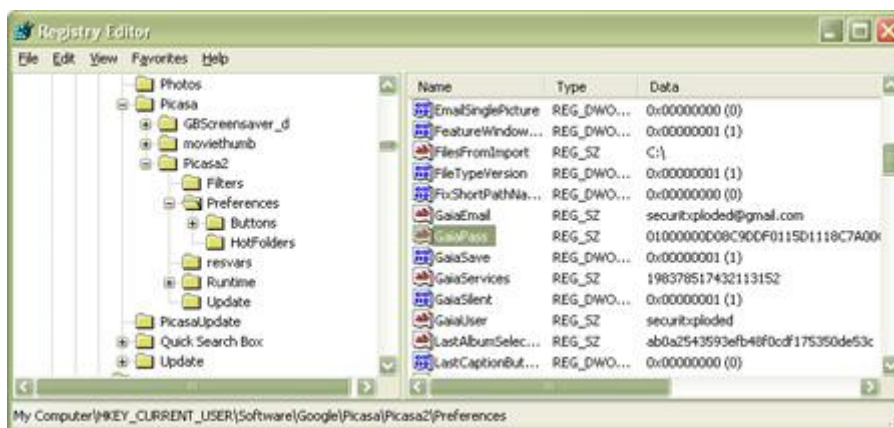
Picasa is the free photo editing software from Google. It facilitates managing and uploading of photo albums to online Google picasa store for sharing with the community. It uses the same Google account for transferring pictures to online web account and stores this password in encrypted format for subsequent logins.

Picasa stores the Google account login details at one of the following location in the registry. However latest version of Picasa (3.6) still uses picasa 2 registry location for storing the settings

```
HKEY_CURRENT_USER\Software\Google\Picasa\Picasa2\Preferences
```

```
HKEY_CURRENT_USER\Software\Google\Picasa\Picasa3\Preferences
```

The Google account name & encrypted password is stored in the registry values '**GaiaUser**' & '**GaiaPass**' respectively as shown below.



Like Google Talk it also uses Windows Cryptography mechanism to protect the password. Here are the different methods used by various versions of Picasa to decrypt the password

Decrypting Picasa 2 (or earlier versions) Password

Here are the basic steps to decrypt the Picasa stored password for previous versions
Retrieve the stored username & encrypted password from above registry location.

Convert the account name to format '**gaia::<account name>**' (for example "gaia::securityxploded")

Modify the encrypted password with crypto magical operations.

Next derive the crypto key using the modified account name as password.

Finally decrypt the password using **CryptDecrypt** function by passing the derived 'crypto key' and

modified password.

Decrypting Picasa 3 Password

Here are simple steps to recover the Google password from latest version of Picasa (Version 3.6)

Retrieve the stored username & encrypted password from above registry location
Convert the encrypted password from hex-string to hex-binary format.
Use the CryptUnprotectData function to decrypt the password in clear text.

Gmail Notifier & Google Password

Gmail Notifier is the standalone systray plugin which notifies user about incoming mails in currently configured gmail account.

Based on Internet Explorer version, Gmail Notifier uses different method to store the google account password. For IE version 7 or later, it uses 'Windows Credential Provider' for securely storing the password. Here are simple steps to recover the password...

Enumerate through all the stored password in '**Windows Credential Provider**' using CredEnumerate function.

Select the entries which are associated with Google account by checking if TargetName begins with text '**Microsoft_WinInet_www.google.com:443**'

For each of these discovered Google accounts, decrypt the password using CryptUnprotectData function.

Here is the sample code illustrating this method.

```
Credits : Thanks to SapporoWorks for original work

void DecryptGmailNotifierPassword()
{
    DATA_BLOB DataIn;
    DATA_BLOB DataOut;
    DATA_BLOB OptionalEntropy;
    tmpSalt[37];
    char *strSalt={"abe2869f-9b47-4cd9-a358-c22904dba7f7"};

    char strURL[1024];
    char strCredentials[1024];
    char strUsername[1024];
    char strPassword[1024];

    //Create the entropy/salt required for decryption...
    for(int i=0; i< 37; i++)
        tmpSalt[i] = (short int)(strSalt[i] * 4);
}
```

```

OptionalEntropy.pbData = (BYTE *)&tmpSalt;
OptionalEntropy.cbData = 74;

DWORD Count;
PCREDENTIAL *Credential;

//Now enumerate all http stored credentials....
if(CredEnumerate(NULL,0,&Count,&Credential))
{
    for(int i=0;i<Count;i++)
    {
        if( (Credential[i]->Type == 1) &&
            _strnicmp(Credential[i]->TargetName, "Microsoft_WinInet_www.google.com",
strlen("Microsoft_WinInet_www.google.com")) == 0 )
        {
            DataIn.pbData = (BYTE *)Credential[i]->CredentialBlob;
            DataIn.cbData = Credential[i]->CredentialBlobSize;

            if(CryptUnprotectData(&DataIn, NULL, &OptionalEntropy, NULL,NULL,0,&DataOut))
            {
                //Extract username & password from credentails (username:password)
                sprintf_s(strCredentials, 1024, "%S", DataOut.pbData);

                char *ptr = strchr(strCredentials, ':');
                *ptr = '\0';
                strcpy_s(strUsername, 1024, strCredentials);
                ptr++;
                strcpy_s(strPassword, 1024, ptr);

                printf("Gmail Notifier Stored account details are, Username=%s,
Password=%s", strUsername, strPassword);

            }
        }
    } // End of FOR loop

    CredFree(Credential);
}

} //End of function

```

Gmail Notifier uses the 'Protected Storage' to store the Google account password for IE version below 7. Here are simple steps to recover such a password.

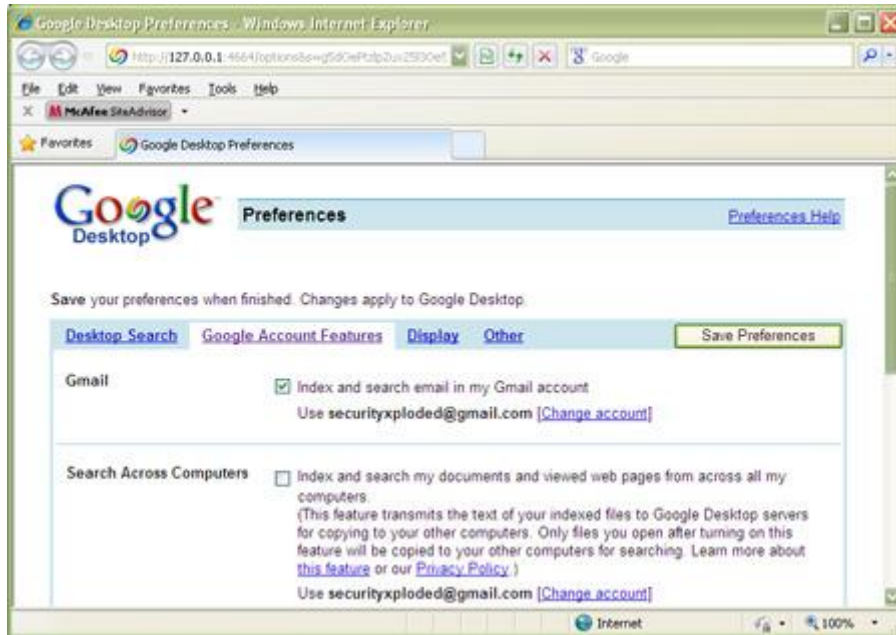
Use the '**Protected Storage**' API functions to enumerate through the stored website entries. Select the entries which are associated with Google by checking if name contains text '**www.google.com**'

Then read the credentials for this account using PStore functions and parse them out.

Gmail notifier is no longer available as standalone application and its now integrated with GTalk.

Revealing Gmail Password from Google Desktop Search

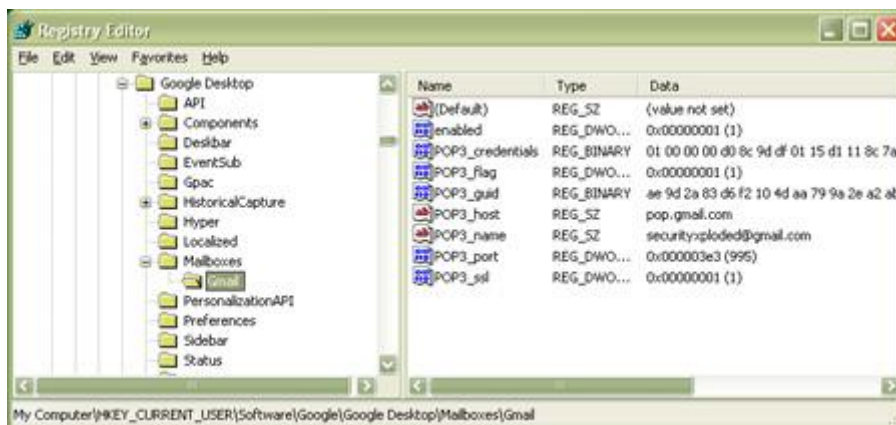
'**Desktop Search**' is Google's solution to searching on the local system. Additionally you can configure it to index & search your Gmail account by setting up the options as shown below.



Any such configured Google account is stored in the following registry location.

```
HKEY_CURRENT_USER\Software\Google\Google Desktop\Mailboxes\Gmail
```

The '**POP3_name**' & '**POP3_credentials**' registry values hold the account name & encrypted password as shown below



Here are steps to recover this password

Retrieve the Google account name & encrypted password from the above mentioned registry location

Next use the **CryptUnprotectData** function to uncover the password in plain text.

Recovering Google Password From Web Browsers

This section explains how each of these popular browsers store the passwords, how to distinguish between Google & other passwords and finally how to recover the Google password from their secret store.

Firefox & Google Password

Firefox stores the account passwords in its sign-on secret store using **Triple-DES** encryption coupled with **BASE64** encoding technique. Different versions of Firefox used different method to store the login passwords. Initial versions of Firefox used **signons.txt** while latest versions uses **signons.sqlite** (SQLite database file) for storing all login details for visited websites.

Firefox stores all website passwords including Google passwords ofcourse at the user consent. To recover the Google password from this big list we need to distinguish between the Google & other passwords.

This task is not difficult as Firefox stores the website URL along with encrypted username & password for each of the stored login entries. Here we just need to check if URL contains the magic string **'google.com'** and then recover only those details to recover real Google username & password.

Internet Explorer & Google Password

Like Firefox and most other browsers, Internet Explorer also stores the sign-on credentials for all visited websites.

Before version 7, Internet Explorer used the famous **'Protected Storage'** to store such sign-on passwords. Since it was less secure and easy to decipher, with version 7 onwards IE uses **'Credential Provider'** store & 'Windows Cryptography' functions to securely store the passwords.

As IE will be storing the passwords for all the websites, we need to separate out Google passwords from it. For older version using **'Protected Storage'** mechanism we can simply check for URL entries against **'google.com'** to get the stored Google login details. However for version 7 onwards we need to have Google login URLs in the IE history database as explained in above research article.

So before we proceed to recover Google Password, we need to add following login URLs

<https://www.google.com/accounts/servicelogin>

<https://www.google.com/accounts/serviceloginauth>

It depends on which URL is used by user to login to Google account. Generally such URLs will be in IE history but sometimes it may have been deleted accidentally by user.

You can use **IEPasswordDecryptor** to add these URLs to IE history database. Once we add these URLs to the IE history we can proceed to recover any stored Google passwords IE Credential store.

Google Chrome & Google Password

Like Internet Explorer and other browsers, **Chrome** also stores the login passwords for all visited websites based on user consent. Chrome uses Sqlite database to store the account information in encrypted format.

Now in order to distinguish between Google & other account passwords we just need to check for '**google.com**' in the URL for each of entries.

Opera & Google Password

Opera browser also stores the login username & password for all visited websites at user's content. Opera uses the DES algorithm to encrypt the password and store it along with other details in the magic wand file.

Each of such stored entries contain the main URL & login URL of the website. Here we have check each of login URL for '**google.com**' to recover only Google account passwords.

Recovering Google Password From Messengers

Most of the universal messengers such as Trillian, Digsby, Paltalk etc supports Google chat as well as other protocols such as Gtalk, Yahoo, AIM etc. Like web browsers these messengers also store the login details including password for future use.

But not all of them store the account passwords locally. Some of them actually store it in their servers. Hence it is difficult to recover such account passwords.

Here we will present details on recovering the login passwords from Messengers such as **Paltalk**,

Pidgin, Miranda etc. These messengers store the passwords locally on user's system in their own encryption format and storage mechanism.

Paltalk Messenger & Google Password

Paltalk is one of the emerging messenger of recent times which supports multiple messenger protocols including Google chat. It stores the login account passwords in the registry using the different encryption mechanism for main and other protocols.

As mentioned in this article, login passwords for each of the protocols are stored in the registry under unique subkey. Google account passwords are stored under subkey named '**GGL**'. So once we find this key, we can decrypt the encrypted password stored under this key to get the Google password.

Pidgin Messenger & Google Password

Pidgin (formerly **GAIM**) is a popular universal messenger which across multiple platforms including Windows & Linux. It supports most of the messenger protocols including aim, msn, yahoo, myspace, msn, windows live, gtalk etc.

Like other messengers Pidgin stores all remembered passwords locally in the file "**Accounts.xml**" at following location.

```
[Windows XP]
C:\Documents and Settings\\Application Data\.purple

[Windows Vista & Windows 7]
C:\Users\\AppData\Roaming\.purple
```

Older versions (Gaim) used **.gaim** folder instead of .purple to store the account details. For each stored account, 'Accounts.xml' file contains the <account> tag, which has sub tags <name> & <password> containing the account email address and password in plain text respectively.

In order to distinguish between Google and other accounts, we need to look at <protocol> field and check if its contains '**Jabber protocol**' as shown below.

```
<protocol>prpl-jabber</protocol>
```

Since Jabber is generic protocol we can cross-check against the account email address and check for "**gmail.com**" to be certain about Google account.

Miranda Messenger & Google Password

Miranda is the new universal messenger which also supports most of popular chat protocols including Google. Miranda stores the login passwords in the local database file using its own proprietary format.

Miranda uses **Jabber** protocol for Google and Gmail chat. As a result all such Jabber based accounts are stored under protocol name '**JABBER**' in its database. Here we need to distinguish Google from other Jabber accounts such as Gmail.

For each Jabber protocol, Miranda stores 'LoginServer', 'LoginName' & 'LoginPassword'. Here we can use '**LoginServer**' as the distinguishing key among different Jabber accounts. For Google accounts, LoginServer is set to '**gmail.com**'. Using this information we can easily recover only Google account passwords from Miranda password store.