

## Internal Storage and Encryption Mechanism

Both MSN/Windows Live Messenger uses Windows built-in '**Credential Store**' to securely store the login account passwords. Not only Windows uses it to store network authentication passwords, but also other applications such as **Outlook, Remote Desktop, Gmail Notifier** etc uses the same mechanism for storing their login passwords. Windows also provides **Credential Management API functions** to allows applications to seamlessly manage this 'Credential Store'.

Windows 'Credential Store' supports different type of password storage mechanisms. Each type uses different kind of encryption and requires different level of privileges for decryption.

Here are the main types

- Generic Password
- Domain Password
- Domain Visible Password / .NET Passport
- Certificates

Though both MSN and Windows Live Messenger uses the same '**Credential Store**' mechanism but they use different types to store the passwords. Here we will see how each of them uses Credential Store to store their secrets and how to recover the stored passwords from it.

## Recovering Password from MSN Messenger

As mentioned MSN Messenger also uses 'Credential Store' to securely store the remembered passwords. These passwords are stored as type '**Domain Visible Password**' aka '**.Net Passport**'. In this 'Domain Visible Password' type only password is encrypted and user name will be stored in clear text.

Here is the complete code sample for recovering and decrypting this type of passwords

```
void EnumerateDotNetPassportPassword()
{
    DATA_BLOB DataIn;
    DATA_BLOB DataOut;
    DATA_BLOB OptionalEntropy;
    tmpSalt[37];
    char *strSalt={"82BD0E67-9FEA-4748-8672-D5EFE5B779B0"};

    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 == CRED_TYPE_DOMAIN_VISIBLE_PASSWORD)
        {
            DataIn.pbData = (BYTE *)Credential[i]->CredentialBlob;
            DataIn.cbData = Credential[i]->CredentialBlobSize;

            sprintf_s(strUsername, 1024, "%S", Credential[i]->UserName);

            if(CryptUnprotectData(&DataIn, NULL,
                                &OptionalEntropy, NULL,NULL,0,&DataOut))
            {
                //Decrypted data contains password in clear text
                sprintf_s(strPassword, 1024, "%S", DataOut.pbData);

                printf(".Net Passport Account details,
                        Username=%s, Password=%s", strUsername, strPassword);

            }

        }

    }

} // End of FOR loop

CredFree(Credential);
}

} //End of function

```

The above code uses the **CredEnumerate** function to go through all the stored network password accounts for current user. Next it checks if the account type is **CRED\_TYPE\_DOMAIN\_VISIBLE\_PASSWORD**. If such an account is found then it decrypts the password data using the **CryptUnprotectData** function. Upon successful decryption it contains the password in clear text.

As this mechanism is used by other applications also, we need to distinguish MSN stored passwords from

other applications. It is not that difficult, here we can just check if the name for each recovered credential entry (Credential->TargetName) matches with text **'.Net Passport'**.

Since it was earlier only MSN Messenger used this technique it also popularly called as **'.Net Passport Method'**

## Recovering Password from Windows Live Messenger & Windows Live Mail

Windows Live Messenger uses **'Credential Store'** to securely store the passwords. All versions of Live Messenger & Windows Live Mail (including **latest 2011 edition**) uses same storage and encryption mechanism to store the credentials.

Here is the sample code which shows how to decrypt the 'Windows Live' password

```
void DecryptWindowsLivePassword()
{
    DWORD Count;
    PCREDENTIAL *Credential;

    char strPassword[1024];

    //Now enumerate all http stored credentials...
    if(CredEnumerate(NULL, 0, &Count, &Credential))
    {
        printf("CredEnumerate found %d accounts", Count);

        for(unsigned int i=0;i<Count;i++)
        {
            printf("Found account %d - %s ", Credential[i]->Type, Credential[i]->TargetName);

            if( strstr(Credential[i]->TargetName, "WindowsLive:name=") )
            {
                printf("Found Windows Live account %d - %s ", Credential[i]->Type,
                Credential[i]->TargetName);

                //convert password to ascii
                strPassword[0]=0;
                WideCharToMultiByte(CP_ACP, 0, (LPCWSTR) Credential[i]->CredentialBlob,
                Credential[i]->CredentialBlobSize/2, strPassword, 1024, NULL, NULL );
                strPassword[Credential[i]->CredentialBlobSize/2]=0;
            }
        }
    }
}
```

```
        printf("Windows Live Account => Username: %s & Password: %s ", Credential[i]-
>UserName, strPassword);
    }

    } //end of for loop

    CredFree(Credential);

}

} //End of function
```

The above code uses the **CredEnumerate** function to go through all the stored network password accounts for current user. Next it checks if the account type is **CRED\_TYPE\_GENERIC**. If generic type of account is found then it decrypts the user credential data using the CryptUnprotectData function which is part of '**Windows Crypto API Package**'. Upon successful decryption it contains both username and password in the clear text separated by semicolon.

Once we recover the stored credentials, we need to check if it belongs to Live Messenger. It stores the passwords with the target name as '**WindowsLive:name=<email\_id>**'. So by checking each recovered entry for 'WindowsLive' text we can get all the login passwords stored by Windows Live Messenger.