Теория и практика восстановления паролей Internet Explorer

Passcape Software

1 Введение

Вряд ли кто будет оспаривать тот факт, что на сегодняшний день **Internet Explorer** является одним из самых популярных обозревателей сети Интернет. По статистике, около 70 процентов всех пользователей сети предпочитают пользоваться именно этой программой. Можно бесконечно спорить о его достоинствах и недостатках, но то, что данный браузер является флагманом своей индустрии - факт, не требующий доказательств. Internet Explorer имеет несколько встроенных технологий, предназначенных облегчить жизнь рядовому пользователю. Одна из них - **IntelliSense** - предназначена для решения рутинных задач. Например, автоматического заполнения адреса посещаемых Web страниц, заполнение полей формы, паролей пользователей и т.д.

Сейчас на многих сайтах необходима процедура регистрации, подразумевающая обязательный ввод имени пользователя и пароля. Если список таких сайтов переваливает за дюжину, то тут уже не обойтись без менеджера паролей. Все современные браузеры имеют в своем арсенале встроенный менеджер паролей, и Internet Explorer не исключение. Действительно, зачем каждый раз запоминать очередной пароль, который в конечном итоге все равно будет забыт. Гораздо проще настроить браузер, чтобы он выполнят всю рутинную работу по запоминанию и хранению паролей за тебя. Удобно и комфортно.

Все бы ничего, да только после неудачной переустановки или краха Windows можно запросто потерять список драгоценных паролей. Вот она - расплата за комфорт и удобство. Хорошо, что практически на каждом сайте есть спасительная кнопка 'Забыл пароль'. Но и она не всегда избавит вас от лишней головной боли.

Каждый производитель программного обеспечения решает проблему с забытыми паролями по-своему. Один официально рекомендует скопировать пару файлов в новый каталог, другой высылает зарегистрированным пользователям специальную утилиту, позволяющую управлять миграцией приватных данных, третий упорно не желает видеть очевидную проблему. Однако спрос рождает предложение. И сейчас в сети программы для восстановления паролей пользуются устойчивым спросом.

Давайте попробуем в этой статье классифицировать типы хранимых в Internet Explorer приватных данных, познакомимся с программами для их восстановления и посмотрим взятые из реальной жизни примеры восстановления утерянных паролей Интернет.

2 Типы хранимых в Internet Explorer паролей

В Internet Explorer могут храниться следующие типы паролей:

- Пароли к Интернет сайтам
- Данные полей автозаполнения
- Пароли автозаполнения
- Пароли FTP

- Пароли синхронизации для кэшированных сайтов
- Пароли удостоверений
- Данные автозаполнения форм
- Пароль Content Advisor

Пройдемся по этому списку чуть более подробно.

2.1 Пароли Интернет

Пароли к Интернет сайтам представляют собой логины и пароли к Web сайтам, обрабатываемые библиотекой **wininet.dll**. Например, при входе в защищенную область какого-либо сайта, может последовать следующий диалог на ввод имени пользователя и пароля (рисунок 1).

Рисунок 1. Диалог ввода Интернет пароля.

Если в этом диалоге установлена опция *'Запомнить пароль'*, то приватные данные будут сохранены на вашем локальном компьютере. Старые версии Windows 9x хранили эти пароли в PWL файле пользователя, Windows 2000 и выше - в защищенном хранилище.

PWL (сокращенно от PassWord List) представляет собой защифрованный файл, типа USERNAME.PWL, где USERNAME - имя пользователя. В PWL файле хранятся RAS пароли, пароли к Web сайтам, учетным записям электронной почты и т.д.

Защищенное хранилище (Protected Storage) системный механизм, обеспечивающий приложения специальным интерфейсом для хранения приватных данных пользователя.

2.2 Данные полей автозаполнения

Данные для полей автозаполнения аналогично хранятся в защищенном хранилище и представляют собой списки из имен полей html формы и соответствующий этим полям информации пользователя. Например, если html страница содержит диалог ввода почтового адреса, то после того, как пользователь ввел свой адрес, в защищенное хранилище пропишется название html поля, сам почтовый адрес и время последнего доступа к нему.

Название html страницы и имя сайта не запоминаются. Хорошо это или плохо? Трудно сказать, скорее да, чем нет. Есть очевидные плюсы: экономится свободное место, увеличивается скорость работы браузера. Если вы думаете, что последняя оговорка несущественна, попробуйте представить что было бы, если бы пришлось делать несколько дополнительных проверок в многотысячном (а это не редкость) списке полей автозаполнения.

Еще один очевидный плюс это то, что данные для одинаковых по названию (и часто по смыслу) полей html формы будут находиться в одном месте и при заполнении этих страниц будут использованы общие данные.

Поясним на примере. Если одна html страница содержит поле автозаполения с названием 'email', и пользователь ввел в это поле свой электронный адрес, то Internet Explorer запишет в свое хранилище, грубо говоря, 'email=my@email.com'. В дельнейшем если этот пользователь зайдет на другой сайт, на котором присутствует страничка с таким же названием поля ввода 'email', ему будет предложено заполнить ее тем значением, которое вы ввели на первой странице (my@email.com). Т.е. браузер обнаруживает в себе как бы небольшие признаки интеллекта.

Главный недостаток такого метода хранения данных вытекает из его достоинства, описанного выше. Представьте себе, что пользователь ввел данные автозаполнения на какой-либо странице. Зная имя поля html формы, можно создать свою простейшую html страницу с аналогичным полем, запустить ее с локального носителя. И тогда, чтобы узнать эти данные, не придется даже подключаться к Интернет и открывать оригинальный WWW адрес.

2.3 Пароли автозаполнения

Однако в случае с паролями, как вы уже могли предположить, автоматической подстановки не произойдет. Поскольку пароли автозаполнения как раз хранятся вместе с именем Web страницы, и каждый пароль привязывается только к одной определенной html странице.

В новой версии **Internet Explorer 7** и пароли, и данные автозаполнения шифруются совсем по иному принципу, избавленному от вышеописанного недостатка. Если это вообще можно назвать недостатком.

Стоит еще отметить, что Internet Explorer позволяет самостоятельно управлять параметрами автозаполнение из меню настроек (рисунок 2).

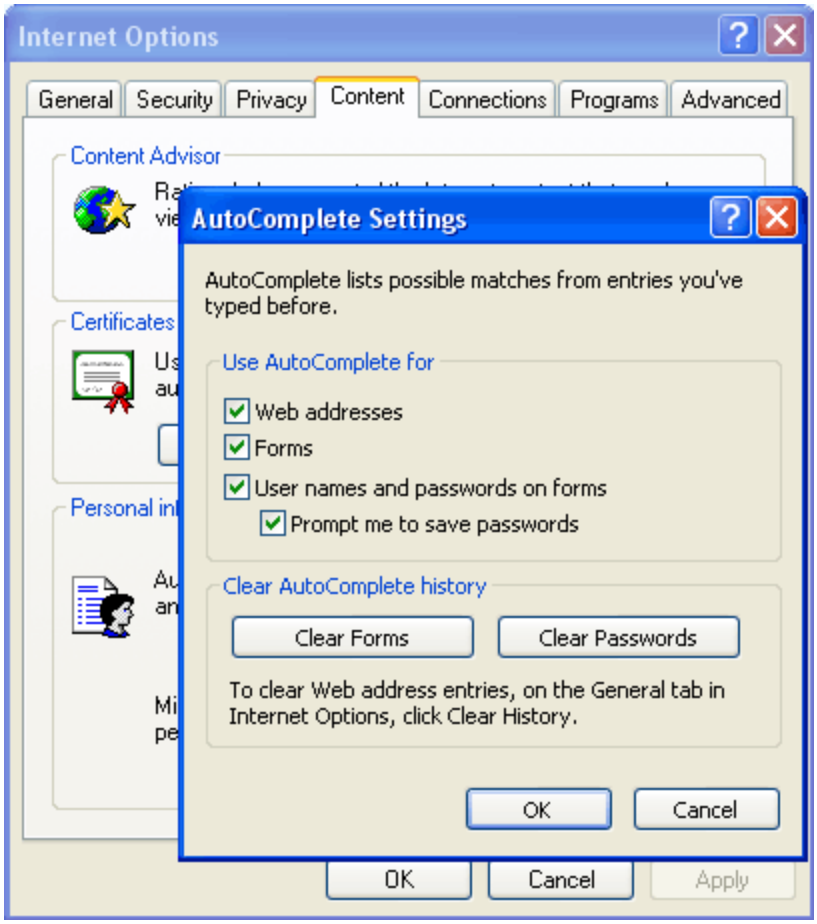

Рисунок 2. Параметры автозаполнения Internet Explorer.

2.4 Пароли FTP

Аналогично работает и механизм сохранения паролей к FTP сайтам. Будет нелишним упомянуть что, начиная с Windows XP, пароли FTP хранятся с использованием дополнительно шифрования **DPAPI**. В нем принимает участие пароль на вход в систему. Естественно, это заметно осложняет задачу ручного восстановления таких утерянных паролей, поскольку теперь дополнительно необходимо наличие **Мастер Ключа** пользователя, знание **SID** и пароля учетной записи.

DPAPI (Data Protection Application Programming Interface) интерфейс, разработанный специально для защиты персональных данных в Windows 2000 и более поздних операционных системах. DPAPI не отвечает за хранение данных, только за их шифрование.

Мастер Ключ (Master Key) - первичный материал, из которого формируются все остальные ключи шифрования.

SID - сокращенно от Security IDentifier.

2.5 Пароли синхронизации

Пароли синхронизации позволяют избавить пользователя от необходимости каждый раз вводить пароль для кэшированных сайтов (сайтов, выбранных для offline доступа). Пароли этого типа тоже хранятся в защищенном хранилище Internet Explorer.

2.6 Пароли удостоверений

Равно как и пароли удостоверений. Механизм разграничение доступа на основе удостоверений не получил широкого распространения в продуктах Microsoft, за исключением, пожалуй, **Outlook Express**.

2.7 Данные автозаполнения форм

Отдельной строкой следует сказать о механизме автозаполнения on-line форм, который представляет собой гибридный вариант хранения данных. При этом сами данные хранятся в Protected Storage, а URL, которому они принадлежат - в реестре пользователя. Записываемый в реестр URL хранится не открытым текстом, а в виде хэша. Вот алгоритм чтения данных автозаполнения форм IE 4 - 6:

```
//Get autoform password by given URL
BOOL CAutoformDecrypter::LoadPasswords(LPCTSTR cszUrl, CStringArray *saPasswords)
{
   assert(cszUrl && saPasswords);

   saPasswords->RemoveAll();

   //Check if autoform passwords are present in registry
   if ( EntryPresent(cszUrl) )
   {
```

```
        //Read PStore autoform passwords
        return PStoreReadAutoformPasswords(cszUrl,saPasswords);
    }

    return FALSE;
}


//Check if autoform passwords are present
BOOL CAutoformDecrypter::EntryPresent(LPCTSTR cszUrl)
{
    assert(cszUrl);

    DWORD dwRet, dwValue, dwSize=sizeof(dwValue);
    LPCTSTR cszHash=GetHash(cszUrl);

    //problems computing the hash
    if ( !cszHash )
        return FALSE;

    //Check the registry
    dwRet=SHGetValue(HKCU,_T("Software\Microsoft\Internet Explorer\IntelliForms\SPW"), cszHash, NULL,
&dwValue, &dwSize);
    delete((LPTSTR)cszHash);

    if ( dwRet==ERROR_SUCCESS )
        return TRUE;

    m_dwLastError=E_NOTFOUND;
    return FALSE;
}


//retrieve hash by given URL text and translate it into hex format
LPCTSTR CAutoformDecrypter::GetHash(LPCTSTR cszUrl)
{
    assert(cszUrl);

    BYTE buf[0x10];
    LPTSTR pRet=NULL;
    int i;

    if ( HashData(cszUrl,buf,sizeof(buf)) )
    {
        //Allocate some space
        pRet=new TCHAR [sizeof(buf) * sizeof(TCHAR) + sizeof(TCHAR)];
        if ( pRet)
        {
            for ( i=0; i<sizeof(buf); i++ )
            {
                // Translate it into human readable format
                pRet[i]=(TCHAR) ((buf[i] & 0x3F) + 0x20);
            }
            pRet[i]=_T('');
        }
        else
```

```
        m_dwLastError=E_OUTOFMEMORY;
    }

    return pRet;
}


//DoHash wrapper
BOOL CAutoformDecrypter::HashData(LPCTSTR cszData, LPBYTE pBuf, DWORD dwBufSize)
{
    assert(cszData && pBuf);

    if ( !cszData || !pBuf )
    {
        m_dwLastError=E_ARG;
        return FALSE;
    }

    DoHash((LPBYTE)cszData,strlen(cszData),pBuf,dwBufSize);
    return TRUE;
}


void CAutoformDecrypter::DoHash(LPBYTE pData, DWORD dwDataSize, LPBYTE pHash, DWORD
dwHashSize)
{
    DWORD dw=dwHashSize, dw2;

    //pre-init loop
    while ( dw-->0 )
        pHash[dw]=(BYTE)dw;

    //actual hashing stuff
    while ( dwDataSize-->0 )
    {
        for ( dw=dwHashSize; dw-->0; )
        {
            //m_pPermTable = permutation table
            pHash[dw]=m_pPermTable[pHash[dw]^pData[dwDataSize]];
        }
    }
}
```

Следующее, седьмое поколение браузера, по всей видимости, собирается сделать этот механизм хранения информации пользователя главным, отказавшись от старого и доброго Protected Storage. Проще говоря, данные и пароли для автозаполнения теперь будут храниться тут.

Что же такого особенного и интересного в этом механизме, что MS решил использовать его в качестве основного? А прежде всего самой идеей шифрования, отнюдь не новой, но простой и гениальной до безобразия. Ее суть состоит в том, чтобы не хранить ключи шифрования данных, а формировать их по мере необходимости. В качестве исходного материала для таких ключей служит Web адрес html страницы.

Давайте посмотрим, как все это реализовано на практике. Ниже представлен упрощенный алгоритм работы IE7 при сохранении данных или паролей автозаполнения:

1. Запоминаем адрес Web страницы. В дальнейшем этот адрес будет использоваться в качестве ключа шифрования (EncryptionKey).
2. Получаем ключ записи RecordKey. RecordKey = SHA(EncryptionKey).
3. Подсчитываем контрольную сумму RecordKey для проверки целостности ключа записи (целостность самих данных нам будет гарантировать DPAPI). RecordKeyCrc = CRC(RecordKey).
4. Шифруем данные ключом шифрования EncryptedData = DPAPI_Encrypt(Data, EncryptionKey).
5. Сохраняем в реестре RecordKeyCrc + RecordKey + EncryptedData.
6. 'Забываем' EncryptionKey.

Без знания оригинального адреса Web страницы, восстановить пароль очень и очень сложно. Расшифровка выглядит довольно тривиально:

1. При посещении оригинальной Web страницы, берем ее адрес (EncryptionKey) и получаем ключ записи RecordKey = SHA(EncryptionKey).
2. Проходим по списку всех ключей записи в поиске RecordKey.
3. Если RecordKey найден, расшифровываем данные, которые хранятся вместе с этим ключом, с помощью EncryptionKey. Data = DPAPI_Decrypt(EncryptedData, EncryptionKey).

Не смотря на свою кажущуюся простоту, этот алгоритм шифрования Web паролей является одним из самых сильных на сегодняшний день. Но у него есть один существенный недостаток (или достоинство, смотря с какой стороны посмотреть). Если изменить или забыть оригинальный адрес Web страницы, то восстановить пароль к ней будет невозможно.

2.8 Пароль Content Advisor

Последний в нашем списке - пароль Content Advisor (пароль-допуск в Русской нотации). Изначально Content Advisor разрабатывался как средство разграничения доступа к сайтам. Но почему-то стал нелюбим многими пользователями (можете с этим не согласиться). Если вы однажды включили Content Advisor, ввели пароль, а потом забыли его, то не получите доступа к большинству сайтов интернета. К счастью или несчастью это легко исправить.

Сам пароль Content Advisor не хранится в явном виде. Вместо этого, подсчитывается его MD5 хэш и записывается в реестре Windows. При проверке, введенный пароль хэшируется и полученный хэш сравнивается с тем, что хранится в реестре. Посмотрите исходный текст проверки пароля Content Advisor, взятый из программы Passcape Internet Explorer Password Recovery:

```
void CContentAdvisorDlg::CheckPassword()
```

```
{
    CRegistry registry;

    //read the registry
    registry.SetKey(HKLM, "SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Ratings");

    BYTE pKey[MD5_DIGESTSIZE], pCheck[MD5_DIGESTSIZE];
    if ( !registry.GetBinaryData("Key",pKey,MD5_DIGESTSIZE) )
    {
        MessageBox(MB_ERR,"Can't read the password.");
        return;
    }

    //Get one set by user
    CString cs;
    m_wndEditPassword.GetWindowText(cs);
    MD5Init();
    MD5Update((LPBYTE)(LPCTSTR)cs,cs.GetLength()+1);
    MD5Final(pCheck);

    //Check hashes
    if ( memcmp(pKey,pCheck,MD5_DIGESTSIZE)==0 )
        MessageBox(MB_OK,"The password is correct!");
    else
        MessageBox(MB_OK,"Wrong password.");
}
```

Первая мысль, которая может возникнуть, это попытаться подобрать пароль путем атаки перебором или по словарю. Однако есть более элегантный способ. Можно просто удалить хэш из реестра. И все... Еще лучше не удалять его совсем, а переименовывать. Чтобы при случае можно было восстановить обратно. Некоторые программы позволяют также делать проверку пароля Content Advisor, вытаскивать подсказку о пароле, включать/отключать его непосредственно и т.д.

3 Программы для восстановления паролей Internet Explorer

К слову сказать, не все программы для восстановления паролей догадываются о таком широком разнообразии. Скорее всего, это связано с тем, что некоторые пароли, например, пароли синхронизации, на практике встречаются достаточно редко. А пароли к FTP сайтам вытянуть довольно непросто. Вот краткий обзор наиболее популярных коммерческих продуктов, восстанавливающих пароли самого полулярного браузера на земле :)

Advanced Internet Explorer Password Recovery от известной фирмы ElcomSoft не распознает пароли AutoForm и зашифрованные пароли FTP. Хотя не исключено, что последняя версия программы уже научилась это делать. Простой, удобный интерфейс. Есть возможность автообновления через Интернет.

Internet Explorer Key от PassWare, аналогично не видит некоторые типы паролей. Программа иногда падает с критической ошибкой при чтении некоторых нестандартных типов ресурсов Internet Explorer. Показывает первые два символа расшифровываемых паролей. Из достоинств следует отметить спартанский интерфейс и удобство работы.

Internet Explorer Password от Thegrideon Software неплох, но может восстанавливать только три типа паролей Internet Explorer (в большинстве случаев этого бывает вполне достаточно). Корректно работает с FTP паролями. Версия 1.1 имеет проблемы с восстановлением паролей автозаполнения. Имеет удобный интерфейс чем-то напоминающий AIEPR. Поражает воображение великолепный по своей красоте и функциональности сайт фирмы.

Internet Password Recovery Toolbox от Rixler Software, немного функциональнее предыдущих. Может восстанавливать зашифрованные ftp пароли, удалять выбранные ресурсы. Однако присутствуют некоторые ошибки программирования. Например, невозможно удалить некоторые типы записей Internet Explorer. К программе прилагается хороший и подробный файл помощи.

ABF Password Recovery от ABF software - неплохая программа, с приятным интерфейсом. Список поддерживаемых типов записей Internet Explorer небольшой. Зато работает она с ними корректно. Относится скорее к области многофункциональных, поскольку может восстанавливать пароли и к другим программам.

Основной недостаток всех вышеперечисленных программ - это возможность восстановления паролей только текущего залогиненного пользователя.

Как уже было сказано, основная масса хранимых в Internet Explorer ресурсов находится в специальном хранилище, которое называется Protected Storage. Protected Storage был специально разработан для хранения персональной информации, поэтому функции для работы с ним (они называются PS API) недокументированны. Впервые Protected Storage был представлен с выходом 4-ой версии IE, которую, к слову сказать, в отличие от третьей версии, писали с нуля.

Так вот, до последнего момента все программы, восстанавливающие пароли Internet Explorer, использовали эти недокументированные API. В связи с чем, накладывалось одно значительное ограничение: PS API могут работать только с паролями текущего, залогиненного пользователя. При шифровании хранимой в Protected Storage информации кроме всего прочего используется SID пользователя, не зная которого практически невозможно (при современном развитии вычислительной способности компьютеров) расшифровать сохраненные пароли.

Protected Storage имеет очень продуманный метод шифрования данных с применением мастер ключей и алгоритмов **des**, **sha-1** и **sha-hmac**. Аналогичные алгоритмы шифрования данных применяются сейчас в большинстве современных браузерах, например, Opera или FireFox. А Microsoft между тем тихим сапом уже разрабатывает и опробывает новые. На момент написания этого текста, в пре-бета-версии Internet Explorer 7, Protected Storage задействован только для хранения паролей FTP.

Из анализа этой предварительной версии становится ясно, что Microsoft готовит очередной 'сюрприз' в виде новых, интересных алгоритмов шифрования. Еще точно неизвестно, но скорее всего при шифровании приватной информации, будет задействована новая фирменная технология защиты данных **InfoCard**.

Так что, можно с большей долей уверенности утверждать, что с выходом Windows Vista и 7ой версии Internet Explorer, пароли будут храниться и шифроваться принципиально новыми алгоритмами. А интерфейс Protected Storage, судя по всему, станет открытым для сторонних разработчиков.

Немного жаль, поскольку, на наш взгляд, истинный потенциал Protected Storage до конца так и не был раскрыт. Постараемся объяснить, почему:

- Во-первых, Protected Storage имеет модульную структуру, с возможностью подключения других провайдеров хранилища. Но за 10 лет существования PS так и не было создано ни одного нового. System Protected Storage - единственный провайдер в операционной системе, используемый по умолчанию.
- Во-вторых, Protected Storage имеет свою, встроенную систему разграничений прав доступа, которая почему-то не используется ни в Internet Explorer, ни в других программах MS.
- В-третьих, не совсем понятно, зачем MS решили отказаться от Protected Storage при хранении паролей и данных автозаполнения. Отказаться именно как от проверенного и надежного механизма хранения, а не шифрования данных. Логично было при внедрении нового алгоритма шифрования, оставить PS как минимум для хранения данных. Наверняка этому есть свои веские причины, поэтому было бы интересно услышать на этот счет мнение специалистов MS.

4 PIEPR - первое знакомство

Программа Passcape Internet Explorer Password Recovery была специально разработана для того, чтобы обойти ограничение PS API и сделать возможным восстановление паролей напрямую из двоичных файлов реестра. Кроме того, в ней имеется ряд дополнительных возможностей для продвинутых пользователей.

Мастер программы предлагает на выбор несколько режимов работы:

Автоматический режим. Расшифровка паролей текущего пользователя посредством обращения к закрытому интерфейсу PS API. Все пароли текущего пользователя, хранящиеся на данный момент в Internet Explorer, расшифровываются одним щелчком мыши.

Ручной режим. Восстановление паролей в обход PS API. Главное преимущество данного метода - возможность расшифровки паролей из вашей старой учетной записи Windows. Для этого необходимо как минимум указать путь к файлу реестра пользователя. Обычно файлы реестра недоступны для чтения, но примененная в программе PIEPR технология, позволяет это делать (при наличии прав локального администратора).

Файл реестра пользователя называется ntuser.dat и находится в его профиле, обычно *%SYSTEMDRIVE%:\Documents and Settings\%USERNAME%*, где *%SYSTEMDRIVE%*, - системный диск с установленной операционной системой, *%USERNAME%* - как правило, имя учетной записи. Например, путь к файлу реестра может выглядеть так: *C:\Documents and Settings\John\ntuser.dat*

Если вы когда-то являлись счастливым обладателем Windows 9x/ME, то после обновления вашей операционной системы на Windows NT, Protected Storage предусмотрительно сохраняет копию ваших старых приватных данных. В результате в Protected Storage могут оказаться несколько идентификаторов пользователя и PIEPR при их расшифровке дополнительно попросит вас выбрать необходимый (рисунок 3).

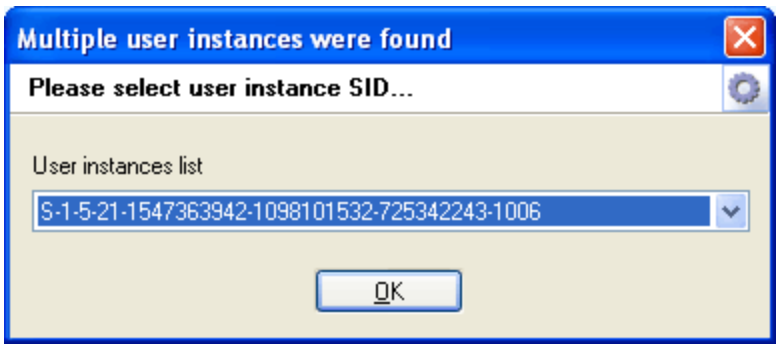

Рисунок 3. Выбор пользователя Protected Storage.

Один из предложенных SID будет содержать данные, оставшиеся после старой Windows 9x/ME. Эта информация дополнительно шифруются логон паролем пользователя, и в настоящий момент их расшифровка программой PIEPR не поддерживается.

Если в ntuser.dat содержатся зашифрованные пароли, например, пароли к FTP сайтам, то программе для их расшифровки потребуется знать дополнительную информацию (рисунок 4):

- Логон пароль пользователя, чьи пароли расшифровываются
- Полный путь к Мастер Ключу пользователя
- SID пользователя

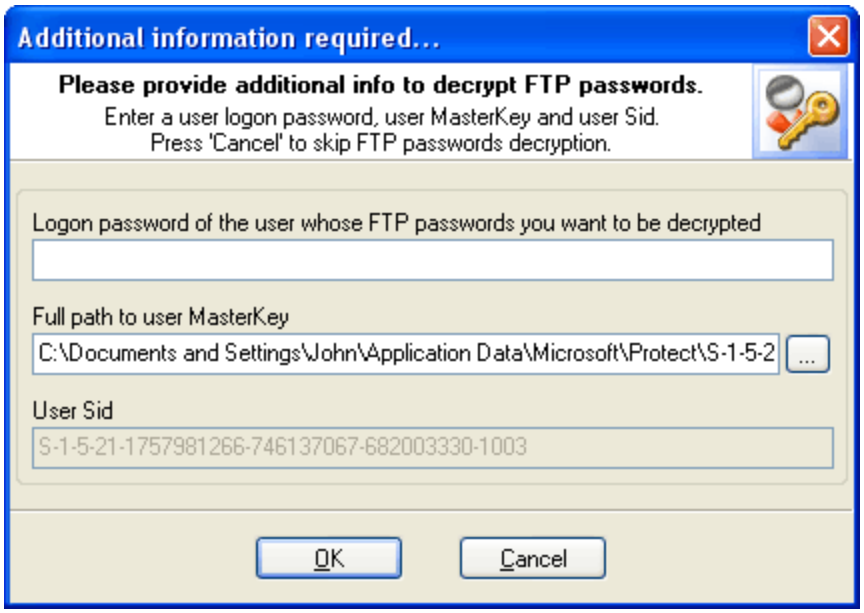

Рисунок 4. Диалог ввода дополнительной информации для расшифровки паролей FTP.

Обычно, последние два пункта программа находит в профиле пользователя и заполняет автоматически. Однако, если ntuser.dat был скопирован из другой операционной системы, то вы должны сами позаботиться об этом.

Самый легкий путь сделать это - скопировать весь каталог с Мастер Ключом (их может быть несколько) пользователя в папку с ntuser.dat. Мастер Ключ находится в следующем каталоге локального компьютера: *%SYSTEMDRIVE%:\Documents and Settings\%USERNAME%\Application Data\Microsoft\Protect\%UserSid%*, где *%SYSTEMDRIVE%*, - системный диск с установленной операционной системой, *%USERNAME%* - имя вашей учетной записи, *%UserSid%* - SID пользователя. Например, путь к каталогу, в котором находится мастер ключ, может выглядеть так: *C:\Documents and Settings\John\Application Data\Microsoft\Protect\ S-1-5-21-1587165142-6173081522-185545743-1001*. Еще раз подчеркиваем, что желательно скопировать целиком весь каталог *S-1-5-21-1587165142-6173081522-185545743-1001*, поскольку в нем может находиться несколько Мастер Ключей. Тогда PIEPR выберет правильный ключ автоматически.

Некоторые папки Windows помечает как скрытие или системные и поэтому они невидимы из проводника Windows. Чтобы они стали доступными, включите в настройках отображения папки возможность показа скрытых и системных объектов, либо используйте альтернативный файл-менеджер.

После того, как каталог с Мастер Ключом пользователя был скопирован в папку с ntuser.dat, PIEPR автоматически найдет необходимые данные и вам останется только ввести пароль пользователя для расшифровки паролей FTP.

Content Advisor. Пароль к Content Advisor, как уже было сказано, не запоминается в явном виде, а хранится в виде хэша. В диалоге управления паролем Content Advisor достаточно всего лишь удалить (вы можете восстановить удаленный пароль позже в любое время) или изменить этот хэш, чтобы разблокировать сайты, заблокированные Content Advisor. PIEPR также покажет вашу подсказку о пароле, если таковая имеется.

Asterisks passwords. Четвертый режим работы мастера PIEPR, в котором можно восстановить пароли IE, прячущиеся за звездочками. Для этого достаточно лишь перетащить лупу на окно с **** паролем. С помощью данного инструмента можно восстанавливать пароли и к другим программам, использующим IE Frames. Например, проводник Windows, некоторые браузеры на базе IE и т.д.

Мы с вами рассмотрели основные режимы восстановления паролей Internet Explorer. Есть еще несколько дополнительных возможностей для просмотра и редактирования кукисов, кэша, истории посещений и т.д. Не будем останавливаться на них подробно, а лучше рассмотрим несколько примеров восстановления паролей с помощью PIEPR.

5 Три примера из реальной жизни

5.1 Восстановление FTP паролей текущего пользователя

При заходе на FTP сайт Internet Explorer открывает диалог авторизации (рисунок 5).

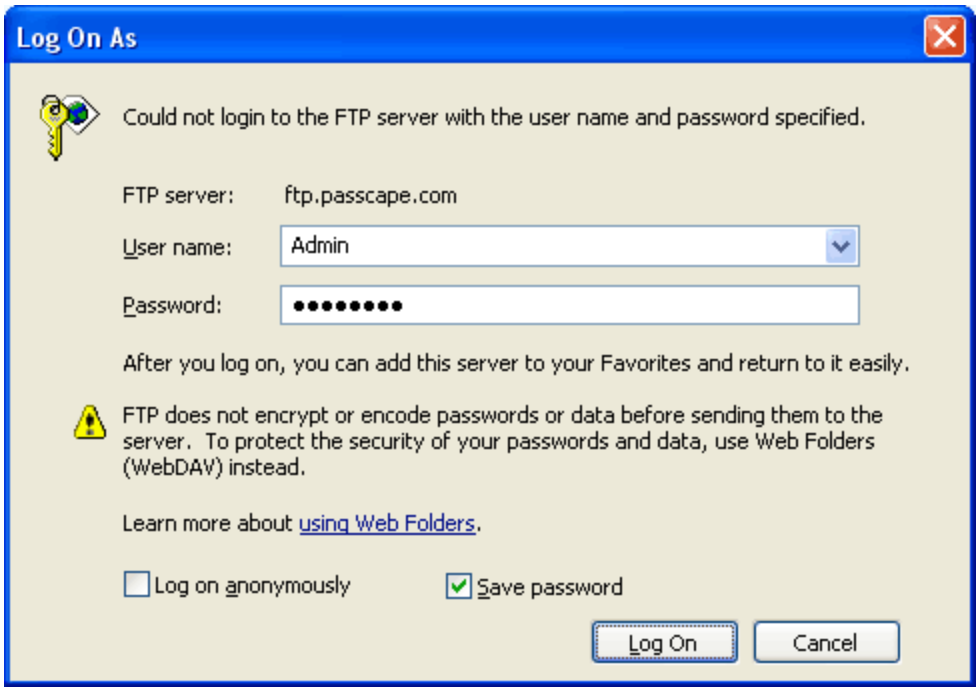

Рисунок 5. Диалог авторизации FTP.

Если вы раньше уже заходили на данный сайт и устанавливали опцию '*Сохранить пароль*', то пароль должен находиться в Protected Storage и восстановление его - задача довольно тривиальная. Выбираем автоматический режим работы мастера PIEPR и нажимаем *'Далее'*. В появившемся диалоге с расшифрованными паролями ищем наш ресурс (имя сайта должно находиться в столбце 'Resource name').

Как видим, расшифровка паролей текущего пользователя не должна вызвать особых затруднений. Да, если пароль вдруг не найден, не забудьте проверить настройки автозаполнения IE (рисунок 2). Возможно, у вас просто не установлена опция сохранения паролей.

5.2 Восстановление паролей Web сайтов из незагружаемой ОС

Типичная, но не смертельная ситуация. Нисколько не реже встречается необходимость восстановления паролей Internet Explorer после неудачной переустановки Windows.

И в том и в другом случае, мы имеем старый профиль пользователя со всеми находящимися в нем файлами. Этого, как правило, бывает достаточно. В случае с переустановкой, Windows предусмотрительно сохраняет старый профиль, переименовывая его. Например, если раньше ваша учетная запись была John, то после переименования она может оказаться *John.WORK-72C39A18*.

Первое и основное, что нам необходимо сделать - получить доступ к имеющимся файлам в старом профиле. Это можно сделать двумя путями:

- На другом винчестере установить новую операционную систему, например, Windows XP, и подключить к ней старый винчестер.
- Сделать загрузочный диск Windows NT. В сети можно найти много разных утилит для создания загрузочных дисков или USB флешек. Можно, например, использовать WinPE или BartPE. Или использовать нашу бесплатную программу Passcape ISO Burner, предназначенную специально для создания загрузочных CD/DVD/USB дисков из ISO образов. Если ваш старый профиль находился на NTFS разделе винчестера, то обязательным условием загрузочного диска должна быть поддержка файловой системы NTFS.

Воспользуемся первым способом. После того, как получен доступ к старому профилю, необходимо разрешить системе показывать скрытые и системные файлы. Иначе необходимые нам файлы будут невидимы. Открываем *Панель управления, Свойства папки*, выбираем вкладку *Вид*. В этой вкладке ищем опцию *'Отображать содержимое системных папок'* и выставляем ее. Сбрасываем опцию *'Скрывать защищенные системные файлы'*. После того, как пароли будут восстановлены, желательно вернуть все на место.

Запускаем мастер программы в ручном режиме и указываем путь к файлу реестра в старом профиле. В нашем случае это *C:\Documents And Settings\John.WORK-72C39A18\ntuser.dat*. Где *John.WORK-72C39A18* - имя старой учетной записи. Жмем *'Далее'*.

Как правило, для расшифровки паролей, имеющейся информации будет достаточно. Однако, если присутствует хотя бы один зашифрованный FTP пароль, то программа

запросит дополнительные данные, без которых расшифровка данного типа паролей не представляется возможной (рисунок 4):

- Пароль пользователя
- Мастер Ключ пользователя
- SID пользователя

Два последние поля, как правило, выставляются автоматически. Если этого по какой-либо причине не произошло, можно сделать наверняка: скопировать в отдельную папку файл ntuser.dat и каталог с мастер ключом. Важно скопировать весь каталог целиком, поскольку в нем могут находиться несколько ключей, и программа сама выберет необходимый. Затем в программе указать путь к скопированному в отдельную папку файлу ntuser.dat.

Все. Теперь нам только остается ввести пароль старой учетной записи, и расшифровка паролей будет завершена. Если вас не интересуют пароли FTP, то диалог ввода пароля пользователя, мастер ключа и SID, можно пропустить.

5.3 Восстановление паролей, хранящихся нетрадиционным способом

Действительно, некоторые сайты 'не дают' браузеру сохранять пароль в списке паролей автозаполнения. Часто такие сайты написаны на JAVA, либо используют альтернативные методы хранения паролей. Например, в кукисах.

Если в поле пароля присутствуют звездочки, то вывод напрашивается сам собой. Выбираем режим работы ASTERISKS PASSWORDS и открываем диалог с 'волшебной лупой'. Затем просто перетаскиваем лупу на окно IE (рисунок 6).

Кукис (cookie) - это текстовый файл, который сохраняется на вашем компьютере Web сервером. Кукисы специально были созданы в помощь пользователям для навигации посещенных сайтов. Однако кукисы часто критикуют за слабый уровень предоставляемой безопасности и неаккуратную идентификацию пользователя.

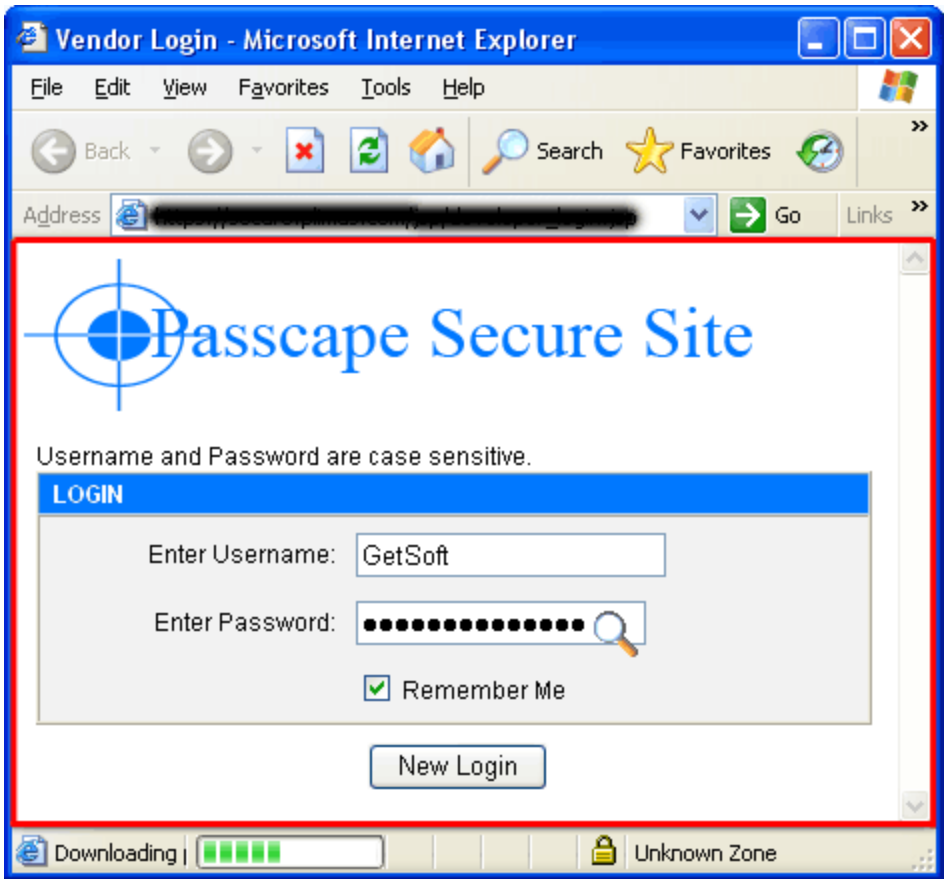

Рисунок 6. Пароль за звездочками.

Пароль (пароли, если в окне IE несколько полей со звездочками) должен появится в окне PIEPR (рисунок 7).

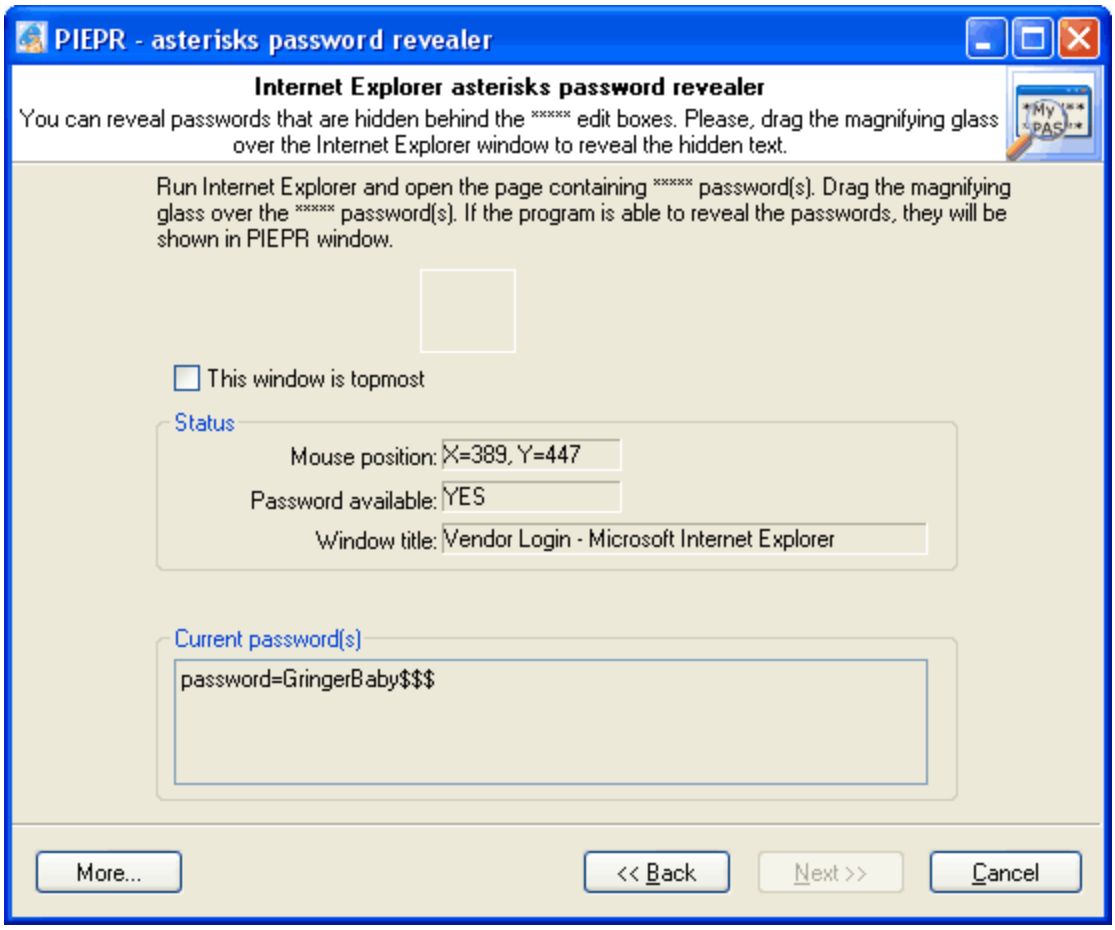

Рисунок 7. Волшебная лупа в действии.

Но не все так просто. Поле для пароля может быть пустым или в этом поле действительно могут находиться ****. В этом случае, как вы уже догадались, инструмент **ASTERISKS PASSWORDS** будет бесполезен.

Если предположить, что пароль хранится в кукисах, то можно попробовать отыскать его. Выбираем инструмент IE Cookie Explorer. В появившемся диалоге появится список сайтов, которые хранят кукисы на вашем компьютере. Кликнем по заголовку столбца URL, чтобы отсортировать список сайтов в алфавитном порядке. Это облегчит нам поиск. Проходим по списку и выбираем необходимый нам сайт. В нижнем списке появятся расшифрованные кукисы для этого сайта (рисунок 9).

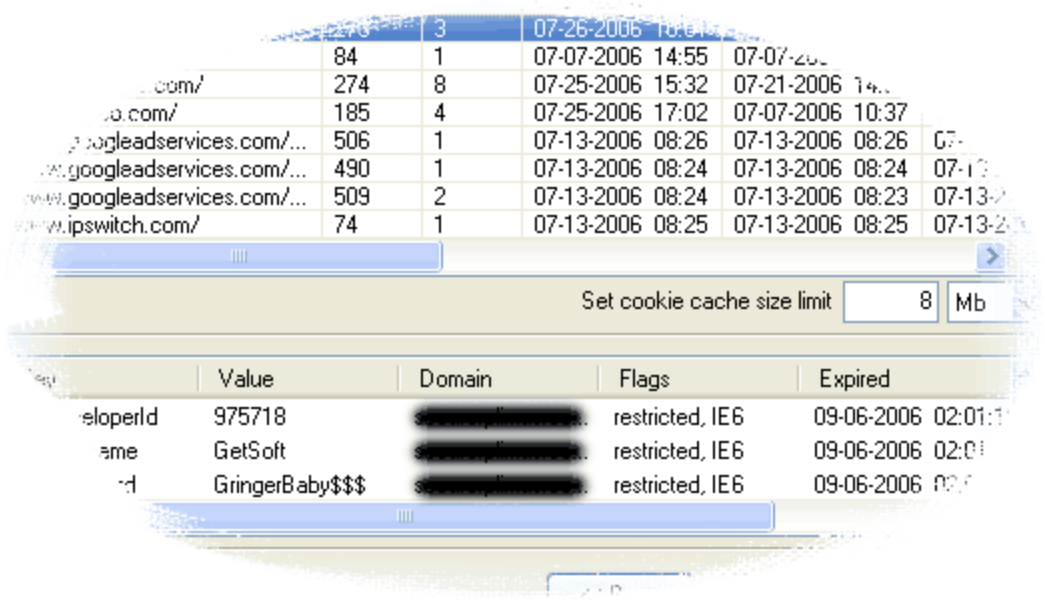

Рисунок 9. Расшифрованные кукисы.

Как видно из рисунка, в нашем случае логин и пароль не зашифрованы и хранятся открытым текстом.

Но довольно часто кукисы бывают зашифрованы, и тогда восстановить пароль у вас, скорее всего, не получится. Единственное, что нам остается, чтобы попытаться восстановить свою старую учетную запись, это создать новую. Тогда можно будет попытаться скопировать в текстовом редакторе старые кукисы и заменить ими новые. Однако это крайний случай и пользоваться им не рекомендуется.

Не забывайте также, что почти на всех страницах и формах с паролями обычно имеется спасительная кнопка '*Забыл пароль*'.

6 Заключение

Как видно из данной статьи, восстановление паролей Internet Explorer - дело довольно простое, не требующее специальных знаний или навыков. Однако, несмотря на кажущуюся простоту, применяемые алгоритмы и схемы шифрования паролей очень хорошо продуманы и реализованы. Хотя самой концепции Protected Storage уже более 10 лет, не забывайте, что она, отлично зарекомендовав себя, была задействована на протяжении трех поколений этого популярного браузера.

С выходом очередной, 7-ой версии Internet Explorer, Microsoft готовит нам принципиально новые схемы защиты приватных данных, применяя улучшенные алгоритмы шифрования и устраняя недостатки, присущие Protected Storage.

В частности, анализ предварительных бета версий Internet Explorer 7 показал, что ключи шифрования паролей автозаполнения теперь не хранятся вместе с данными. Они вообще нигде не хранятся! Это небольшое ноу-хау, которое по достоинству должно быть оценено как профессионалами, так и конечными пользователями, которые в конечном итоге только выиграют от этого.

Но самое главное, будет устранен основной недостаток, присущий Protected Storage - возможность расшифровки паролей без знания дополнительной информации. Проще говоря, потенциальному хакеру было достаточно одного физического доступа к содержимому жесткого диска, чтобы расшифровать хранимые в IE пароли и другие приватные данные пользователя. С выходом Internet Explorer 7 ситуация несколько изменится.

А пока нам только остается с нетерпением ждать выхода Windows Vista и IE7, чтобы подробнее рассмотреть новые механизмы шифрования, применяемые в следующем поколении этого популярного браузера.