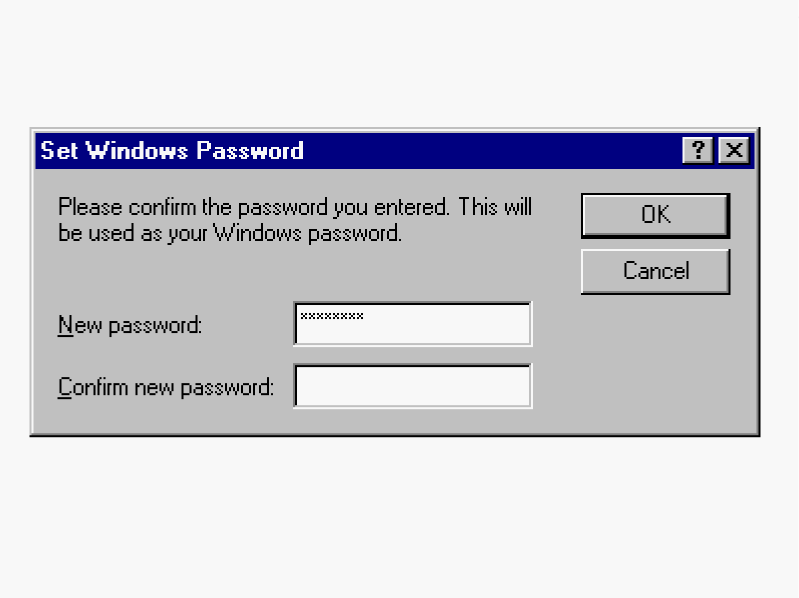
PWL Files

The Achilles' Heel of Windows 9X Client Networks

By Scott D. Winters - 9/14/00

How many times have you logged on to a network through a Windows 95/98 client for the first time, and were greeted with this message:



Set Windows Password

Please confirm the password you entered. This will be used as your Windows password.

New password: ********

Confirm new password:

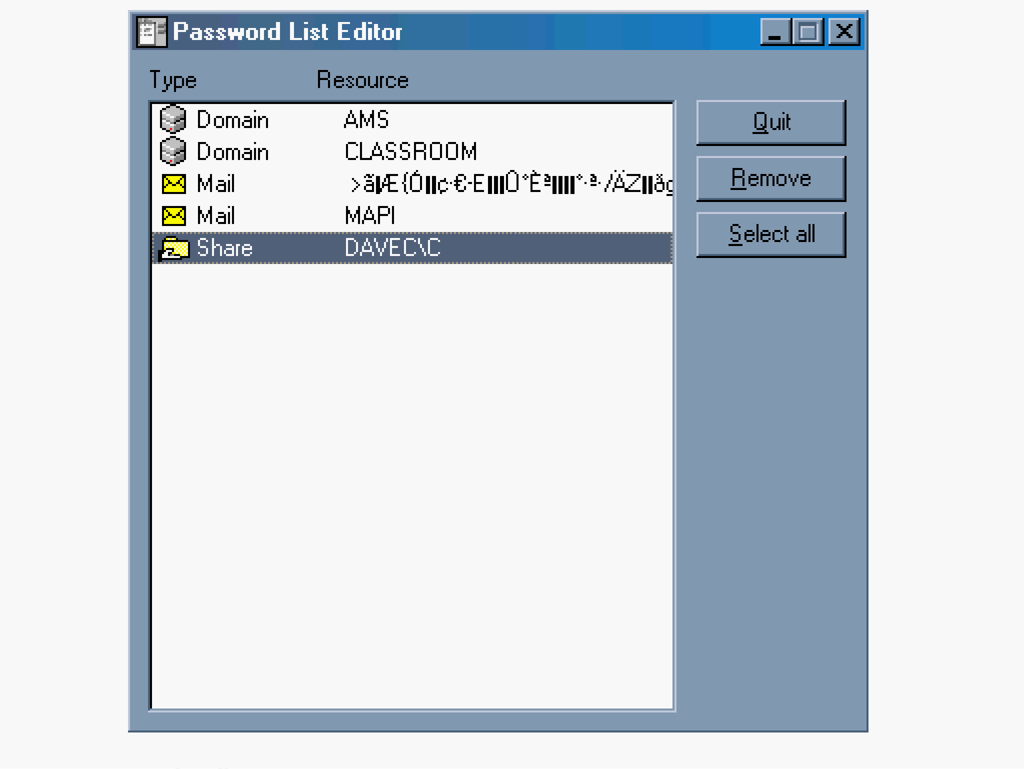
Why does this happen? The server you are logging into *already* knows your password. This is actually a function of Windows itself. Windows creates a file with the extension PWL, which stands for *password list* to hold this information. The file is saved in the directory that Windows 9X (referring to any flavor of Windows 95 or 98) was loaded in, typically C:\WINDOWS. The file's name is the first eight characters of your login name, unless there is already a like named PWL file on the system. This would happen if your logon name was RichardSmith, and RichardSimon had already logged on the system in question. The result would be truncating what would have been a "RichardS" file name to its first five letters - "Richa", and appending 000 to the end, creating a file named Richa000.pwl. If another login occurred by someone whose login name's first eight characters also matched RichardS, then their PWL file would become Richa001.pwl, and so the process would continue.

What purpose does the PWL file serve? What information does the PWL file hold and why? Microsoft does not offer much detailed technical information on the PWL file, because they believe in "*security through obscurity*". This theory says that the less you know about a products inner security workings, the safer it will be. That may hold true if the product is secure to begin with. Typically the most secure systems are proven secure through the testing of their inner-workings by security aficionados. If everyone knows how your product works, and it still isn't defeated, then it is a truly secure product!

Due to Microsoft's lack of "documentation", most technical information that can be found on PWL files is the result of programmer's and security experts' efforts to document the files themselves. Much of this information is through reverse engineering, and experimentation.

The PWL file's function is to hold any cached password information. This is a convenience for the end-user, so they don't have to type in all of those annoying passwords every time they access a passworded resource. The resource could be a share on a neighboring Windows 9X machine, access to a server in an alternate domain, contact to Samba shares on a Unix system, your dial-up networking dialer, or even contact with a Novell NetWare file-server! Any of these could be accessed from a Windows 9X station, and would have the potential of being "cached" in a PWL file.

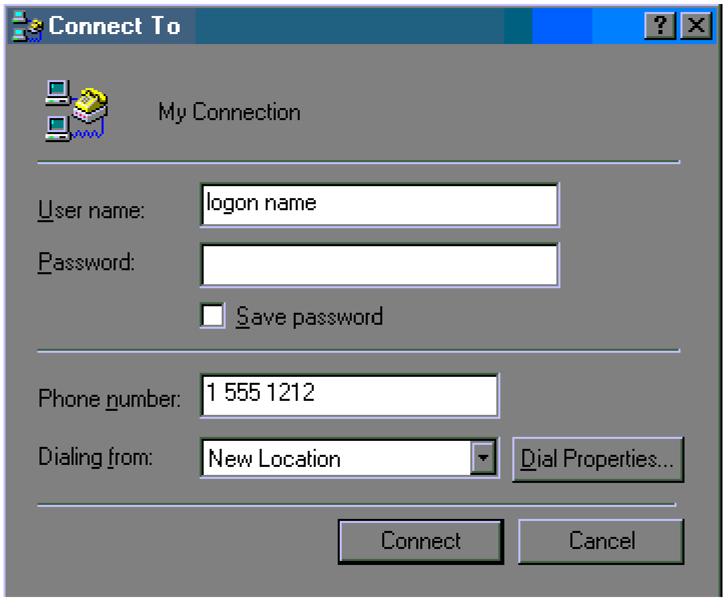
Technically the PWL file is actually a database file. It contains a series of records representing the name of the resource that you are connecting to, the type of resource, and of course the password that enables you to connect to said resource, encrypted of course. Note, each of the three records are defined by the program doing the saving, so the records, in some cases, could be juggled around or used completely differently then mentioned. For a more graphical representation of a PWL file, look at this screen capture of Pwledit.



Type	Resource
Domain	AMS
Domain	CLASSROOM
Mail	>ãÆ{Ó▮▮¢·€·E▮▮▮Ö°Èª▮▮▮▮°·ª·/ÄZ▮▮ăg
Mail	MAPI
Share	DAVEC\C

Pwledit is a tool that comes on the Windows 95/98 CD-ROM. On the Windows 98 CD-ROM find it under the **\tools\reskit\netadmin\pwledit** directory. On the Windows 95 CD-ROM find it under the **\admin\apptools\pwledit** directory. It shows all of the resources that are stored in the currently logged-in user's PWL file with their type represented by a graphical icon and name. The password is the only of the record types not represented. This utility is Microsoft's answer to making PWL files manageable. It allows the removal of individual resources from the PWL file with the remove button (as seen above). Otherwise, the only way you could clear a resource would be to completely delete the PWL file, and recreate it without the resource in question (not a pleasant proposition on a file with many resources!). There are two limits on the resources listed in the PWL. The first is that there can only be a total of 255 total entries. The second is that the service/client that allows login to the resource must be compatible with the PWL format, which at this point means that it must be a Microsoft client.

The entries actually get in the file through the cooperation of the end-user. Like in the example mentioned above, during initial logon when you are asked to confirm your password, Windows is making a PWL for your logon name! By clicking check boxes like the one in Dial-up Networking's dialer that says, "Save Password" (*see below*), you are adding resources to your PWL!



My Connection

User name: logon name

Password:

□ Save password

Phone number: 1 555 1212

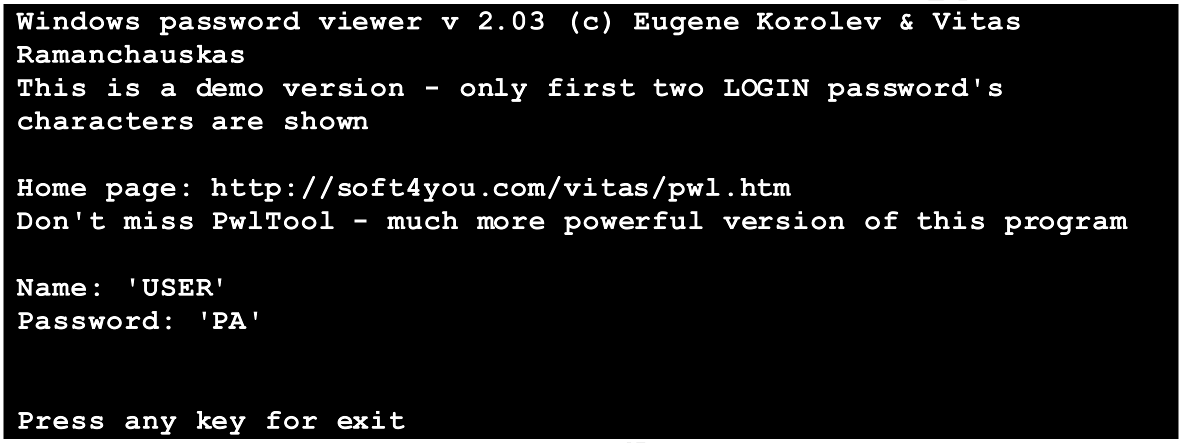
Dialing from: New Location

This isn't such a bad thing, right? PWL files make it easy to logon to a collection of networks with just one logon on startup! Who wants to type in a long, confusing password supplied by your Internet Service Provider every time you want to connect to the Internet? For these reasons PWL files are used quite heavily, despite the fact that they have a history of being, and still are, rather insecure.

What are the security issues of the PWL file? Well, it doesn't take a security guru to determine that one small file filled with lots of network passwords could be a bane to network security. Place this same file on a very insecure operating system (like Windows), and the problems multiply. It's not just a Microsoft network problem, either! Any resource that you allow a Microsoft Windows 9X client to connect to using a Microsoft client will be at risk as well! So Samba shares on Unix and Linux are at risk, as are Novell NetWare Servers that are logged-on to through Microsoft's client for NetWare Networks (if you are using NetWare's client, password's aren't saved in the PWL file!)

There are two main attacks on the PWL file. One is an attack on a user's PWL file that is currently "unlocked", meaning that the user is currently logged on to the local system where their PWL resides. The other is an attack on "locked" PWL files, meaning on the PWL file itself without the necessity of anyone accessing it. No one has to be logged in; you just need to have access to the PWL file you want to decrypt.

The first is the easier, faster, and more successful of the two. It should almost always succeed. All you need is a logged in user, and access to their station. Programs such as Cain 2.0 (available at http://packetstorm.securify.com/trojans/Cain20.EXE) and Pwlview (available at http://soft4you.com./vitas/pwlview.htm) can be used for this attack. It works like this: an attacker walks over to your station when you leave it unattended. They run the attack program (possibly from a floppy disk or shared network drive) which puts the list of cached files up on the screen. *The following example is from the demo version of PWLview 2.0 that only shows the first characters of the user's password.*

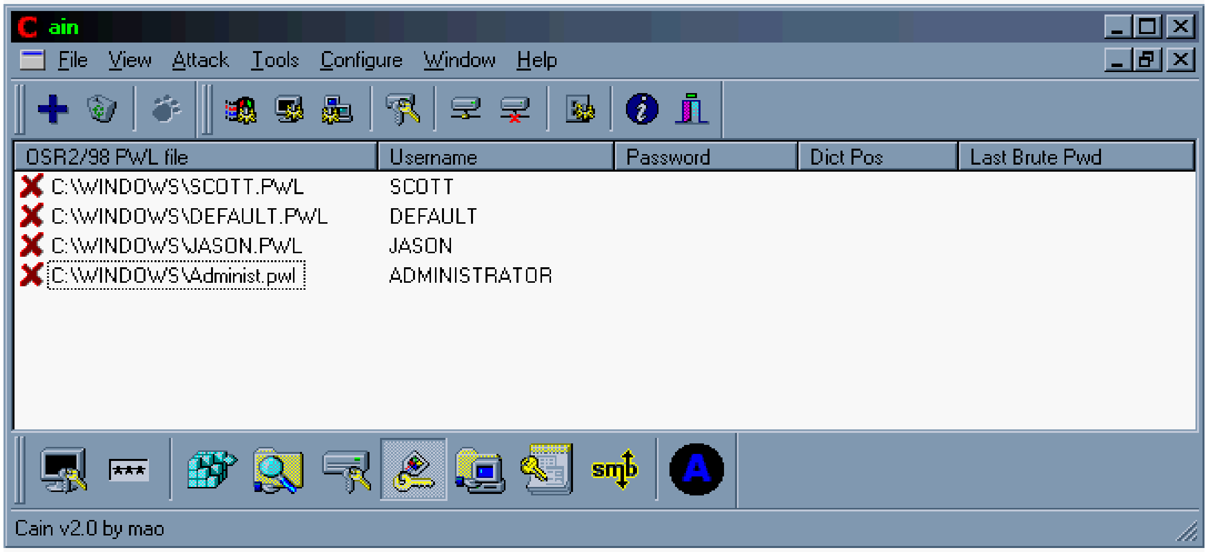


```
Windows password viewer v 2.03 (c) Eugene Korolev & Vitas
Ramanchauskas
This is a demo version - only first two LOGIN password's
characters are shown

Home page: http://soft4you.com/vitas/pwl.htm
Don't miss PwlTool - much more powerful version of this program

Name: 'USER'
Password: 'PA'


Press any key for exit
```

This information can be captured to a text file for reasons of stealth. The attacker saves the file to their floppy disk and walks away with all of your cached passwords! For a serious reality check download one of these programs and load it on a Windows 9x station that *you* use. Be amazed as you see your most secret of passwords in plain text, before your very eyes! The program works by using an undocumented Windows API (*Application Program Interface – meaning a routine pre-programmed by Microsoft facilitating easier advanced programming)* call that reveals the cached passwords. Showing this to *any* user should make them take station level security seriously! Realize there are major limits to this attack. First the user has to be logged on. It can only be run locally, so the user has to leave their desk, *and* remain logged in. Finally the attacker has to be able to get to the user's desk without drawing suspicion. If any or all of these conditions could occur in your environment be concerned! This attack is fast, and it works!

The second type of attack is directed at the files themselves, decrypting the passwords using popular attack methodology. With the first release of Windows 95 (and Windows for Workgroups before it…) the encryption scheme for the PWL files was flawed. This made the decryption of said files easy. A program to do so, called Glide was produced, but it had some flaws in its decryption algorithm, so it often failed. Other PWL cracking programs followed, along with a patch from Microsoft (Microsoft article #Q132807) to "enhance" the PWL file's encryption. It increased the key used for encryption from 32 bits to 128 bits. The larger the key you use for encryption the more difficult the result will be to decrypt (according to Microsoft this change made decryption approximately 8 followed by 27 zeros times harder!!!).

Windows 95 OSR2 and 98 versions incorporated this encryption scheme using the RC4 algorithm and a 128-bit key, which did (as Microsoft said) make it more difficult to decrypt PWL files…but not impossible! A program such as Cain 2.0 (shown below) can be used, which incorporates popular decryption techniques to “crack” the password.



OSR2/98 PWL file	Username	Password	Dict Pos	Last Brute Pwd
C:\WINDOWS\SCOTT.PWL	SCOTT			
C:\WINDOWS\DEFAULT.PWL	DEFAULT			
C:\WINDOWS\JASON.PWL	JASON			
C:\WINDOWS\Administ.pwl	ADMINISTRATOR			

Note: Windows 9X uses the user logon name to generate the Encryption key. Therefore, you must know the user's login to be able to successfully decrypt a PWL file. Unfortunately the name of the PWL file is most often the login name.

There are two main decryption attack styles: dictionary, and brute-force. Dictionary attacks use a file containing hundreds or even thousands of words, including ones popularly used for passwords. This word list can be customized for different languages, or areas. Each word is encrypted in the same way as the PWL file’s password. The resultant encrypted text and the encrypted password from the PWL file are compared. If the two match you have discovered the password!

The second type of decryption attack is called brute-force. It simply goes though every letter/number/special character combination, starting with minimum password size and going up! Otherwise it works similarly to the dictionary attack, taking each combination of characters, encrypting them like the PWL file’s stored password and comparing the two. This attack should always work, however it may take a VERY long time. The longer the password and the more types of characters used (letters, numbers, special characters) the longer the decryption will take. It takes minutes to go through all combinations of a three character word, while it takes HOURS (even days!) to go through all combinations of a seven character word. Selecting good passwords makes your PWL files considerably safer.

How do you protect yourself, knowing that all of these tools exist to steal passwords from your Windows 9X station? There are ways to make your environment safer from each type of attack.

To protect yourself from either attack the key is securing your station. If you want to completely remove this concern you can upgrade your workstations to Windows NT or Windows 2000 since they have increased local security, and don't even use PWL files (however there are tools to crack their SAM password database…but that's another paper).

Another means to remove the PWL file as a problem is by disabling password caching. This was originally the suggested fix when the Windows 95 PWL security issue was discovered, and many security conscious administrators make it a part of their standard Windows 9X station implementation. This can be done in several ways including using the Cain 2.0 tool. A less involved means is to manually add an entry to your registry. Using Regedit add the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Network\DisablePwdCaching =1

Making sure that the **DisablePwdCaching** value is added as a **Dword** value. For more information on this procedure check http://support.microsoft.com/support/kb/articles/q140/5/57.asp or http://www.software.com.pl/newarchive/mailingl/Bugtraq/bugtraq/1995_4/0139.html. After setting the system to disable password caching, delete all PWL files from the system's hard drive and they will not be created again.

Perhaps an easier way to prevent the saving of passwords at your local station may be to simply ***not*** save them when prompted! When you get prompted for a confirm password, or save password check box, simply don't comply. The frustrating part is that you will be prompted every time you login with that annoying little confirm password box, and you'll always have to type your password in for shares and dial-up networking. No matter what, I would suggest a policy like this for Administrator/Admin/Supervisor logon ID's. Their passwords should NEVER be cached on a station's local drive. You may want to go as far as to regularly audit stations to verify that there are no PWL files for these logins on your user's local hard drives. This can be done manually, with a batch file, or even setup in Microsoft Scheduler to be performed once every week or so.

If the inconveniences of typing in user passwords are too great, you can leave PWL files enabled and focus on the local security of your Windows 9X system itself. Always log out or have a means to lock your station when leaving it unattended. Passworded screensavers or third party products allowing you to lock your station are recommended. However none of these solutions prevent someone from rebooting and gaining local access to your station. With local access the attacker can't perform an attack through the API call, since you are not logged-in. They could, however, copy your PWL files off of your local drive and perform a dictionary or brute-force attack against them at their leisure.

To do so, as previously mentioned, they would need your logon name. One way to complicate this process is by having user names greater than eight characters. However this is hardly foolproof and shouldn't be relied on. For example with a PWL file named billsmit.pwl a logon name of billsmith may easily be guessed. If the actual name is billsmithee however, it may provide additional protection. Trying to implement such a logon name policy would be difficult. Also getting users to type in long login names often doesn't go over well.

A better approach to local security may be to defend your station from unauthorized reboot. One way to achieve this is to enable a BIOS password. These passwords are set in the system's hardware so you can't even gain access to your operating system until this password is entered. It is dangerous to rely on such passwords, because some BIOS versions have "backdoors" meaning special alternate passwords that will work no matter what you set the main password to. Also BIOS passwords can be cleared if the attacker can gain access to the inside of your system's case (securing your case with a lock may be a way to defend against such activity.)

Other things you can do to prevent unauthorized access to your system, is disabling floppy and CD-ROM booting. Also requiring an authorized network login to gain access to your local system can be a good security step. This can be done with System Policy Editor a tool available on the Windows CD-ROM. *For more information on Policy Editor check out*: http://support.microsoft.com/support/kb/articles/Q147/3/81.asp

Another important thing to keep in mind is the network security of your Windows 9X system. Passwording shares, and properly securing network accessible software (such as personal web or FTP servers) is vital to safeguard local resources. Often users are under the false conception that there is nothing of value on their system, so securing the system isn't important. However access to PWL files could mean the compromise of the *entire* network.

Finally, if all else fails and attackers do manage to get your PWL files, your last means of defense is using strong passwords. Well chosen passwords help protect you from the threat of decryption attacks. A good password policy demands the use of long passwords (greater than 6 characters was a standard, I would suggest at least 8 - longer is even better!), and passwords that use a combination of letters, numbers, and special characters (!@#$%^, etc.). You should also avoid passwords that can be found in a dictionary or dictionary words with one or two numbers appended on the end. By following all of these rules you make the decrypting of your PWL files VERY difficult.

Hopefully this information has opened your eyes to a major security concern on Windows 9X stations. By auditing your risk, being aware of your exposure, and defending against it by implementing the aforementioned techniques, you can rest easy knowing you have maximized the protection of your station and your network.

References:

Ramanchauskas, Vitas "PWL files" 1997 URL:http://soft4you.com/vitas/pwl.htm (5/16/00)

Microsoft "Enhanced Encryption for Windows 95 Password Cache" Microsoft Knowledgebase 7/27/2000 URL:http://support.microsoft.com/support/kb/articles/Q132/8/07.asp

Graves, Rich "RE:Cracked:WINDOWS.PWL [most services accessed by any version]." Bugtraq Archives 1995 URL: http://www.software.com.pl/newarchive/mailingl/Bugtraq/bugtraq/1995_4/0139.html

Microsoft "Microsoft Windows 95 Password List Security Issue" Microsoft Knowledgebase URL:http://support.microsoft.com/support/kb/articles/Q140/5/57.asp