Global Information Assurance Certification Paper

Password Auditing and Password Filtering to Improve Network Security

Tina MacGregor. Security Essentials GSEC Practical Assignment V1.2d.

Introduction

Passwords in one form or another have become part of our daily lives, whether it is the key code to our home alarm system, the P.I.N. number for the ATM, Internet access, or the network logon at the office. Passwords add a layer of protection to our personal and business information. They are very often the first line of defense and, in many cases, the only line of defense against intrusion. However, most people dislike passwords – they do not like having to remember different passwords for different things, especially if they are required to change passwords at regular intervals. Left to their own devices, most people will choose a password that is short, simple, easy to remember, and obvious. It will invariably be a dictionary word, a name (theirs), or a number like a birthday or social security number – something they already have memorized. Most people do not realize that if the password is simple and obvious to them it is probably going to be just as simple and obvious to a hacker.

So why do so many people object to protecting themselves and their company's assets? In many cases it is simply the lack of awareness of the threats and of the ease and frequency of hacker attacks. SANS has user ID's and passwords as number 8 on their list of Top Ten Security Threats. The more people are made aware of just how easy it is to crack their passwords and gain access to their private lives or business assets, the more they will buy into and therefore adhere to a strong password policy. Without this understanding, the reaction to the Security or IT manager's efforts to improve corporate security is negative or undermined resulting in passwords being written on post-its stuck on the user's monitor or keyboard.

One good way to demonstrate the need for stronger passwords is to show people how easily theirs can be cracked. Many people have no idea that password-cracking tools are readily available, easy to use, and, in many cases, free. The effect on these people can be very dramatic. All passwords can be cracked sooner or later and it is a matter of making them complicated enough that the average hacker gives up or it takes so long to crack the password that the password has changed before it is cracked.

Many networks have a password policy in place. This is configured through User Manager – Policies – Account on NT networks. The NT 4.0 default setting is very weak.

A stronger policy, recommended by many security experts, is that passwords should be at least 6 characters long; mixed upper and lower case letters; and contain at least one numeral. The password should not contain any form of the users name or ID and should not be a dictionary word. A password cannot be reused for 10 password changes.

However, like all security policies, password policies need reviewed and updated in keeping with the increasing sophistication of hackers and their tools. So what may have been a secure policy at one time may no longer be secure in the present environment.

The logon process and weaknesses that make password-cracking easy.

When a user logs on to a NT Network, the user ID and Domain name are sent in clear text to the server. The password itself is not sent. Instead a challenge-response protocol is used which sends an encrypted hash of the password. If this hash matches the hash and user ID stored in the user accounts database on the server then security access tokens are created and the user is authenticated.

There are some weaknesses in NT networks that make password cracking easier than in other networks. In order for Window 98/95 clients to be connected to the network, passwords are stored in two forms: A LAN Manager storage scheme and an NT storage scheme. During authentication both hashes are sent across the network.

The LAN Manager scheme is a weak link. The passwords are non-case sensitive. The password is converted into uppercase, truncated to 14 characters and split into two 7 character pieces. The result is padded, reversed and hashed using DES. For every password that is 7 characters or less the last 7 characters of the 14 character hash are the same. This makes it easier for a password-cracking tool to identify 7 character passwords. The password cracker now has only to crack a 7 character password or if the password is more than 7 characters, one 7 character password and one smaller (the remaining characters) password.

The NT storage scheme is stronger and is encrypted using the MD4 algorithm. Passwords are case sensitive (unless sent from a non-NT client). The passwords are not split up and no salt is added unless NTLMv2 has been enabled. A salt is a random string added to the password before it is encrypted. If a salt is added to a password then if there are identical passwords in the database they will have different hashes and so are unique. By default NT uses NTLMv1, which has no salt. With Service pack 4 came NTLMv2, which does use a salt. However NTLMv2 has to be enabled

in the registry for this to happen. This is enabled in HKEY_ LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa, Value Name: LMCompatibilityLevel. For instructions and registry values refer to Alex Park's paper "Password and Network Logon Security in Windows NT 4.0".

One way to ensure that the password policy is kept up to date is through regular auditing using one of the readily available password crackers. Many people feel that auditing is not enough and that relying on auditing alone to strengthen password policy will be leaving the network open to weak passwords since auditing is done after the weak password has been made and not before. Password filtering, on the other hand, is designed to prevent the weak password from being created in the first place. The password filter is there to ensure that the password policy is adhered to. So, the filter is only as good as the strength of your password policy – the password cracker can help determine what that password policy should be and enable to review and update this policy as needed so that the password filter can do its job.

Auditing passwords using a password cracker.

The strength of passwords can be audited using password-cracking software. There are many password-cracking tools available on the Internet. Some are free.

Before downloading and trying out this or any other password-cracking software, it is essential to obtain written permission from management. Having permission is what separates the security professional from a hacker. Before beginning any password audit, it is important to try and bring management on board by discussing the whole process and what is expected to be achieved. Make sure the written permission includes statements about confidentiality, the security measures to protect the cracked password file, and that the file will be destroyed after the audit and reporting process. It should also state that after the audit and review all passwords will be changed. Similar statements are included in the standard agreement issued by professional password recovery companies like Password Crackers INC at www.pwcrack.com. Including these kinds of statements makes management more comfortable with the process, shows a professional approach, and demonstrates the serious nature of password security.

For the password audit discussed in this paper, the recently released version 3.0 of L0phtCrack was used. L0phtCrack 3.0 is a powerful and sophisticated NT password-auditing tool. Version 3.0 has a number of important enhancements over the previous 2.5 version. These include support for Windows 2000, listing of audit time to crack each password,

distributed cracking which allows auditing to run simultaneous auditing on multiple machines to spread load, and enhanced editing features. This program can be downloaded from www.securitysoftwaretech.com. A 15-day free trial is offered that allows testing of all the features except the brute force crack. After 15 days the program must be registered to go on using it.

How L0phtCrack works

L0phtCrack must first obtain the password hashes. This can be done in one of four ways.

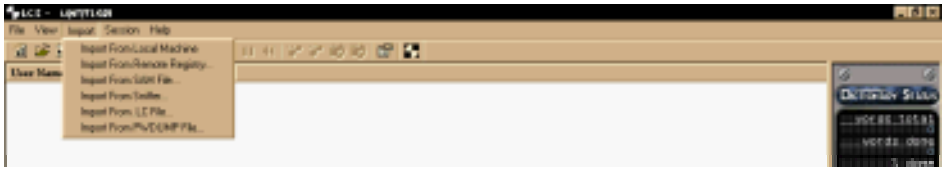
1. With administrator rights the "Import from the local machine" command can be used in the Import menu. This will retrieve the hashes from the local NT or Windows 2000 machine. (Fig.1)

2. Password hashes can be dumped from a remote machine using the "Import remote Registry" from the Import menu. If the remote machine has SYSKEY protection, the hashes obtained using this method will not be cracked. SYSKEY adds an extra layer of encryption to the passwords and is used by default in Windows 2000 and may be enabled on some NT machines (needs SP3). SYSKEY uses a 128-bit random key to encrypt the SAM file. The random key is then encrypted with another key called the System Key. Without the System Key it is impossible to decrypt the SAM file. The PWDUMP3 utility can be used to dump passwords from a remote system that is SYSKEY protected. PWDUMP3 does this by reading directly from memory on the remote system bypassing the SYSKEY encryption. This utility can be downloaded from www.ebiz-tech.com.

3. Password hashes can be obtained by importing a SAM file into L0phtCrack from an emergency repair disk, from the repair file on the hard drive, or a backup tape. This only works on systems that do not use Active Directory.

4.The fourth method is to use the SMB packet capture by running the "Import from Sniffer" command in the Import menu. This captures the encrypted hashes from the network

Figure 1: Import commands in the Import menu on L0phtCrack 3.0.

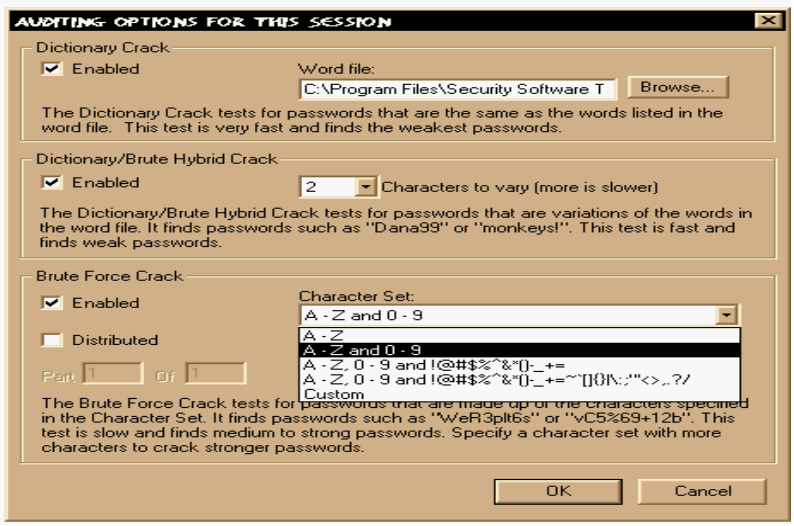


In the audit conducted for this paper, the SAM file was copied from an emergency repair disk from the PDC. If L0phtCrack is being run on an NT machine the SAM file can be imported directly into the program using the Import SAM file command in the Import menu. If it is running on a win 98/95 machine the SAM file needs to be expanded first using the command C:> expand –r A:\sam._ C:\temp. This will expand the SAM file from the disk and put in a directory of choice, in this case the temp folder. The expanded SAM file can then be imported into L0phtCrack as before.

Once the password file has been imported into L0phtCrack, session options can be configured from the Session menu. By default the following is enabled. (Fig2)

1. Dictionary crack using the default words-english file that comes with L0phtCrack. This file has 250,000 words. This default word file can be edited and custom words can be added if needed. This feature is valuable where people work in specialized areas and may tend to use passwords connected with their work. Third party word file can also be imported into L0phtcrack.
2. Hybrid crack adding 2 different characters to a word. The number of characters can be varied from 1 – 13. Most people vary their passwords by adding characters to the end of the word. L0phtcrack checks for characters appended to words.
3. Brute force crack uses every combination of characters it is configured to use. The default character set A-Z 0-9. More complex character sets can be selected from the list or custom character sets created. The more complex the character set the longer the crack will take.

Figure 2. Session options in L0phtCrack 3.0

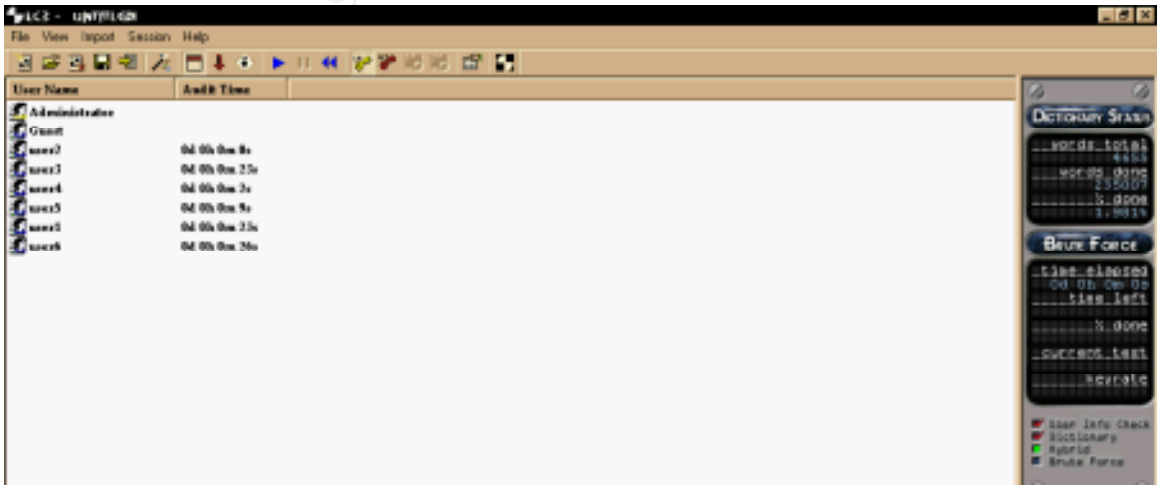


The brute force crack will eventually crack all passwords but may take a long time to do it depending on the complexity of the password.

In the password audit discussed in this paper the original password policy was: passwords should be a minimum of 5 characters; passwords are changed every 90 days; a history of 5 passwords was kept; lock out after 3 bad logon attempts. Out of 172 password hashes imported into L0phtCrack 40% were cracked in less than 10 minutes, 60% in less than 1 hour, 78% in 12 hours and 98% in 72 hours. Only 3 passwords remain uncracked at the time of writing this paper. The results of this audit showed that the password policy needed strengthening.

In order to present this convincingly to management, a password-cracking demonstration was arranged. First, six of the key user accounts were selected including those from upper management. In a test environment using a Windows NT computer these six accounts were created. The passwords used were the original passwords collected from the first audit. An emergency repair disk was made and the SAM file imported into L0phtCrack and an audit carried out. All of the passwords were cracked in less than 30 seconds. In the figure below user names changed for security purposes and the cracked password hidden. L0phtCrack 3.0 allows cracked passwords to be hidden from view but the time to crack them is given so that an administrator knows when they have been cracked. (Fig.3) It is worth noting that the first password cracked by L0phtCrack is the one that is the same as the user ID. The first thing L0phtCrack does in an audit session is check user information before the dictionary attack.

Figure 3 Result of the first password auditing demonstration.



Next, these passwords were altered based on a more secure policy. The passwords were increased to a minimum of seven characters, at least one number and one punctuation character were used in the body of the password and upper and lower case mixed. So that a password like taxman became T^x$Man or banker became B@n1<er. The passwords for the six key accounts were changed to the stronger passwords. An emergency repair disk was made and the SAM file from this imported into L0phtCrack and the audit run. It took L0phtCrack 5 days to crack the weakest of these passwords and all were cracked in 12 days. Although these new passwords are not the strongest, it served as useful demonstration on how an already memorized password could be altered to improve security.

At the management meeting where the results of the password audit were presented. L0phtCrack was run, first with the first password file created on the test computer. People watched in astonishment as their passwords were cracked in less than 30 seconds. This was a fairly convincing demonstration that the present password policy was inadequate in terms of security and resulted in management's complete co-operation and support in changing this policy. The altered passwords were shown and L0phtCrack was run using the second SAM file containing these passwords. This was left running through the rest of the meeting at which point not one character of any password had been cracked. The file of the completed crack was shown with the audit times. This showed a substantial improvement in password strength compared to the original passwords.

Enforcing a stronger password policy.

By default Windows NT 4.0 and Windows 2000 password policy is weak. Blank passwords are permitted, passwords expire in 42 days, changes are allowed immediately, no password history is kept, and there is no account lockout following bad logon attempts. This password policy can be made stronger by configuring it in User Manager – policy – account. This policy affects all users across the domain. Password length, age, uniqueness, lockout after bad logon attempts, lockout duration and reset, and remote user disconnect can all be configured here. However, this will not prevent the use of dictionary words nor enforce the use of non-alpha characters so the strength of this policy is limited

Using password filters is a way of enforcing strong passwords when a password has to be changed - the aim being to catch and prevent weak passwords before they get in to the system.

Stronger password filtering can be enforced using the password filtering utility that came with Win NT Service Pack 2, passfilt.dll. Passfilt.dll is not automatically enabled. It is enabled by adding the value PASSFILT in the Registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages on the PDC and all BDCs. The computers need to be rebooted before password filtering is enabled. Passfilt enforces a minimum of 6 characters in the password that contains 3 of the following; uppercase letters; lower case letters; numbers; and a non-alphanumeric character. The password cannot be any part of your user ID or full name. This filter does not apply to passwords created in User Manager. Passfilt can improve the strength of the passwords chosen but it does not prevent the use of dictionary words or the natural tendency for people to add a number or symbol to the end of a word, both of which are easily cracked.

There are a number of third party "smart password filtering" programs available such as Rainbow Diamond's Password Defender (www.brd.ie), T P Information Systems Password Policy Enforcer (www.securiteam.com) and Mdd Inc.'s Password Bouncer 1.0. (www.mddinc.com).

In this study Password Defender was evaluated. A 21-day evaluation copy of Password Defender can be downloaded from www.brd.ie.

Password Defender can be installed on any standalone system to protect local logons or on a domain controller to protect domain logons. It has three main components that work together and once configured requires little user intervention. These are automatic filtering, auditing and countermeasures.

1. Automatic filtering – prevents passwords from being used that do not comply with the selected policy. Password Defender ships with 5 preconfigured policies ranging from weak to very secure. These can also be easily customized as needed. It also allows for different policies to be applied to different groups of users or individual users. This way more secure policies can be applied to privileged users. The first level policy is the standard NT policy and it is the weakest. The next level policy is equivalent to enabling passfilt.dll. The third level policy is called baseline and is the default for Password Defender. (Fig. 4) The fourth and fifth level policies are called enhanced and maximum and apply much stronger levels of security. (Table 1) By default, all except the NT SP2 policy are configured to filter password changes at the User Manager level. All of these policies can easily be customized to apply fine grain security levels as needed.

2. Automatic auditing – automatic password auditing is carried on in the background when the system is idle. Dictionary, hybrid, and brute force attacks are carried out similar to L0phtcrack. Password Defender uses a dictionary file of 1.2 million words. Auditing and filtering are integrated so that any cracked password is automatically added to the filter list so that it cannot be re-used.
3. Automatic countermeasures – when auditing detects a weak password it can be configured to force a password change, disable the account, or filter on expiry.

FIG 3 Password Defender's Baseline policy.

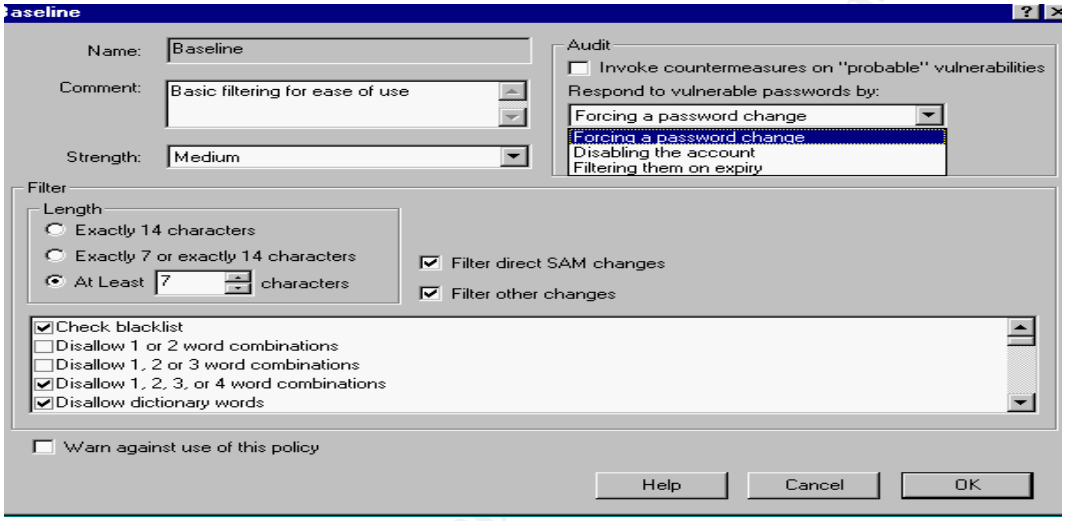
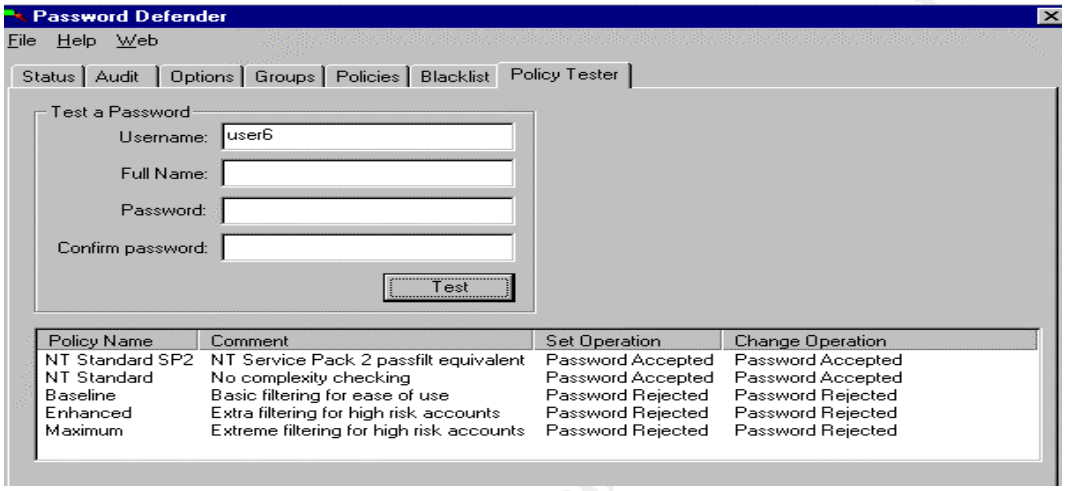


TABLE 1

Policy	NT Standard	NT SP2	Baseline	Enhanced	Maximum
Strength	Vweak	Weak	Medium	High	Very high
Length	0	≥6	≥7	≥13	≥14
Expire password (days)	0	0	90	60	30
Check blacklist	-	-	√	√	√
Disallow 1,2 word combo	-	-	-	-	-
Disallow 1,2,3 word combo	-	-	-	-	-
Disallow 1,2,3,4 word combo	-	-	√	√	√
Disallow dictionary words	-	-	√	√	√
Disallow dictionary words with digit prefixes	-	-	√	-	-
Disallow dictionary words with digit suffixes	-	-	√	-	-
Disallow dictionary words with embedded digits	-	-	-	-	-
Disallow dictionary words with embedded non-alphabetics	-	-	-	-	-
Disallow dictionary words with embedded specials	-	-	-	-	-
Disallow dictionary words with leading, trailing or embedded digits	-	-	-	-	-
Disallow dictionary words with leading, trailing or embedded non-alphabetics	-	-	-	√	√
Disallow dictionary words with leading, trailing or embedded specials	-	-	-	-	-
Disallow dictionary words with punctuation prefixes	-	-	√	-	-
Disallow dictionary words with punctuation suffixes	-	-	√	-	-
Disallow digit prefixes	-	-	-	-	-
Disallow digit suffixes	-	-	-	-	-
Disallow digit/letter substitutions	-	-	√	√	√
Disallow full name sub-strings	-	√	√	√	√
Disallow passwords cracked by auditing	-	-	√	√	√
Disallow punctuation prefixes	-	-	-	-	-
Disallow punctuation suffixes	-	-	-	-	-
Disallow username sub-strings	-	√	√	√	√
Require at least 3 character classes	-	√	√	√	√
Require non-ASCII	-	-	-	-	√
Require some alphabetic characters	-	-	-	√	√
Require some digits	-	-	-	√	√
Require some lower case	-	-	-	√	√
Require some punctuation	-	-	-	√	√
Require some upper case	-	-	-	√	√
Require upper/lower case mix	-	-	-	√	√
Use secondary Passfilt.dll	-	-	-	-	-
Respond to vulnerable passwords by filtering on expiry	√	√	-	-	-
Respond to vulnerable passwords by forcing password change	-	-	√	√	√
Filter direct SAM changes	√	-	√	√	√

In addition, Password Defender has a feature allowing passwords to be added to a blacklist. These could be passwords that, although complex, have been compromised in the past. By default the baseline, enhanced and maximum policies check the blacklist. There is also a policy tester function that allows password to be tested against the different policies. (Fig. 5)

Figure 5: Password Policy Tester



After presentation of the password auditing demonstrations and a thorough review of the findings of this paper a much stronger password policy is being selected, and will be enforced using Password Defender. Periodic reviews of the password policy will be carried out by password auditing using L0phtCrack. Changes will be made as necessary.

Summary

Passwords are often the first line of defense (in some cases the only line of defense) in a network environment or standalone system. Software specifically designed to crack passwords and break through this important line of defense are freely available and easy to use. Passwords need to be complex enough to increase the time that it takes to crack them but not so complex that users forget or write them down in obvious places. But, how complex is complex enough? A combination of password auditing, using real-world auditing tools, and password filtering can help determine what the password policy should be and enforce it. Password-cracking tools are continually improving and developing in sophistication. In light of this, periodic password auditing using these tools should be carried out and the password policy changed as necessary. The password policy should then be enforced using password filtering. In a Corporate environment a live demonstration of password auditing and password filtering tools to upper level management IS a very valuable and effective method of gaining support for enforcing stronger password policies.

References

Chaddock, M. Mary. "A Breakdown of SANS Top Ten Threats". 11 October, 2000. URL http://www.sans.org/infosecFAQ/threats/top_ten.htm

Daily, Sean. "NT Server Security Checklist". July, 1998 URL http://www.ntmag.com/Articles/Index.cfm?ArticleID=3571&pg=4.

Donovan, Craig. "Strong Passwords". 2 June, 2000. URL http://www.sans.org/infosecFAQ/policy/password.htm.

Jumes, G. James, Cooper, F. Neil, Chamoun, Paula, and Feinman, M. Todd. Windows NT4.0 Security, Audit, and Control. Microsoft Press. 1999. 123 – 131.

O'Dwyer, Frank. "Ensuring Password Quality on NT Networks" Revison 90. 7April, 1999. URL http://www.brd.ie/ntsecurity/password.pdf.

Park, Alex. "Password and Network Logon Security in NT 4.0. A Brief Overview of the Windows NT Security Model". 13 June, 2000. URL http://www,sans.org/infosecFAQ/win/logon.htm.

Password Cracker Inc. "Standard Agreement". URL http://www.pwcrack.com/agree.htm.

Sans Institute, "How to Eliminate The Ten Most Critical Internet Security Threats The Experts Consensus". 8 September, 2000 URL http://www.sans.org/topten.htm.

Savill, John. "How do I Enable Strong Password Filtering". 22 December 1999. URL http://www.windows2000faq.com/Articles/Index.cfm?ArticleID=14766.

Security Software Technologies. "What's New in LC3". URL http://www.securitysoftwaretech.com/lc3/whatsnew.html.

Williams, Jim. "L0phtCrack – How Good are Your Passwords?". 3 May, 1999. URL http://netsecurity.about.com/compute/netsecurity/library/weekly/aa050399.htm.