Global Information Assurance Certification Paper

Marsha Williams

SANS Security Essentials

GSEC Practical Assignment, Version 1.4b, Option 2

June 16, 2003

Adventures in implementing a strong password policy

Abstract

Password authentication is high in the list of potential security vulnerabilities. In the October 2001 SANS/FBI Top 20 list [10], absent or inadequate passwords placed second. The revised SANS Top 20 list, first appearing in October 2002 [11], moved password vulnerabilities down to seventh place but they were still a major potential systems risk. This case study relates our experiences in strengthening our password policy.

Passwords turned out to be only a starting point. We effectively strengthened our overall policy but we also learned that sometimes strong password policies and practices combining with human factors can interfere not just with convenience, but with actual usability and needed access.

This paper explores the issues we had to negotiate in strengthening our passwords, some of the of the special situations which had to be handled as exceptions to the policy, and our planned future directions.

Our setting

Our company is a utility, providing services to residences and businesses. Our customer base is large and physically widespread. We have multiple distributed offices housing a total of about 275 employees. The Information Technology (IT) functions are centralized and located in the corporate office. The network uses Windows NT servers with Windows NT/2000 workstations. Access to customer information, maps of our service area, and current weather information are critical to both daily operations and emergency response. IT services must be available 24/7.

Before

Password policies

There was no formal written company password policy at the start of this project, but there existed the software policies set in the Windows NT User Manager for Domains. Passwords were required on all domain and workstation accounts. Other requirements:

- Maximum password age: Passwords expired in 120 days
- Minimum password length: 6 characters
- Minimum password age: 1 day
- Password uniqueness: remember 10 passwords
- Account lockout: none

Administrator accounts

Administrator-level accounts were an exception to the expiration policy. They were set not to expire. Some had not had a password change in more than a year.

User education pro*g*ram

IT provided a low-key user education program. Emails were sent out every few months that stressed the need for password security and gave some tips for good passwords. The tips were generally along the lines of avoiding personal information when choosing a password.

Password auditing

We had no password auditing program.

Security training

Before this project none of the IT staff had had any systematic security training. We did not know about the many tools available and believed that we had few options to increase password security, other than changing administrator passwords more often (though we seldom seemed to get around to doing this) and occasionally reminding our employees not to use their families or favorite sports when picking a password. Then I attended an Instructor-led Online Training (ILOT) version of the SANS GIAC course and learned that there had been a better way to increase password security ever since Windows NT Service Pack 2 had been released – passfilt.dll.

Assembling a password security toolkit

Passfilt.dll

Passfilt.dll is a Windows NT/2000 extension that enforces stronger passwords. It is available if Windows NT is at SP2 or higher, and is also available in Windows 2000, but it is not a default implementation. It is implemented by modifying a registry entry [2].

After passfilt.dll is implemented, passwords:

- Must be at least 6 characters in length

- Must use 3 out of these 4 character types: lower-case letters, upper-case letters, numbers, and non-alphanumberic characters (e.g. punctuation marks)
- Cannot use any part of the individual name or username in the password

Dumpsec

I learned about the Somarsoft utility DumpSec [4] (formerly Dumpacl) through the SANS GSEC course. It has been invaluable for our password and account auditing, especially the "Dump Users as Table" report. This report gives a user's last logon date and the date the user changed a password. If the user has never changed the account password, the word "Never" will appear – it doesn't count if an administrator made the password assignment.

LC3/LC4

IT already knew about the password cracking program LC3 [4] and we had tried it after it was recommended by a security consultant. Our very first demo audit, even before the SANS training, showed us that we might have a problem with our passwords. Formerly known as L0phtcrack, LC3 was upgraded to the newest version, LC4, during the project.

Implementing the change

Writing a password policy

The company had several IT-related policies, mostly relating to acceptable use and the Internet. There was no formal policy on either accounts or passwords. These topics had been discussed from time to time as an area of concern and I had been given the task of educating users on how to make them more secure. As mentioned in the Before section, our user education program was non-specific and very low-key.

At the beginning of the project, I wrote a proposed formal password policy, basing it on the password policy samples in the Sans Security Policy Project [5], and submitted it to our company's process for reviewing and approving policies. It has not at this point completed the approval process. This is just as well because we will be changing this first policy based on our experiences with the project.

Obtaining administrative support

In addition to implementing passfilt.dll, we wanted to make two other very visible changes in our password policy. We needed to get management approval and support first.

The visible proposed changes were:

- requiring longer passwords (moving from 6 to 8 characters)
- requiring complex passwords (using passfilt.dll)
- implementing account lockout for the first time

It took the IT manager about a month to obtain management approval for these changes. Reservations centered upon how usable the longer and more-complex passwords would be for our employees as opposed to the simpler ones we were accustomed to using. We finally received approval after a top administrator experimented with the proposed password rules for her own password and concluded they were do-able.

Marketing the new password policy

Once we had approval, the next step was marketing the upcoming changes to the users. We planned to change *all* passwords if possible, even the administrative ones (it was past time to change these anyhow). Management asked us to give our employees a week's notice to prepare for the change. We did the following:

- The IT team set the changeover date for a Friday evening after the regular work day was over. We had learned earlier that if a password was expired while the user was logged in, Exchange would immediately stop working. Although someone is always on duty in our company, an after-hours change would inconvenience the fewest possible users.
- I sent out email to all employees explaining the change in password length and complexity and giving one or two suggestions on how to create strong passwords. We ultimately provided 4 suggestions, which are briefly described in the "Strong but memorable passwords" section later in this paper.
- I contacted everyone who had access to administrator-level accounts and passwords and informed them that these too would be changed the next week and would expire regularly in the future.
- I contacted two key vendors who had non-expiring passwords, to inform them of the new policy on their accounts. They were to change their own accounts. We would not force the change in this case, since we did not know if the passwords were tied to services which might stop.
- I created and posted a document on the company intranet with a full explanation of some of the suggestions for ways to create strong but memorable passwords.

The first email was sent out a full week ahead of the change. This was followed up during the week with two more messages. Every e-mail included a link to the password instruction page on the company intranet.

Making passwords strong but memorable

In my emails and on the intranet, I gave our employees multiple suggestions for creating strong but memorable passwords. The goal was to create passwords which would not be easily cracked by LC3 or other automated programs but which could be remembered by their owners. The suggestions were derived from multiple internet sources. We adapted most of our techniques from an excellent article on passwords by George J. Kalis [3].

Our suggestions to the employees were to:

- Use a memorable word – it can even be a dictionary word or name – but move the hands up a row from the home row on the keyboard when typing it. This way, “GoFishing” would become “T9R8wy8ht”. This technique would be most usable by touch-typists.
- Create a passphrase and use the first letter of each word. The phrase “Now is the time for all good persons ...” would yield the password “NittfaGp”. Since our rules required still more complexity, I suggested putting a punctuation character in front – “!” for example, to make it “!Nittfagp”.
- Transform words using by substituting characters for letters - @ or ^ for “a”, $ for “s”, 3 for “e”. The word “Geekspeak” might become “G33k$p3^k.”
- Do the unexpected with characters and numbers and put them at the beginning or middle of a password instead of the end. LC3 can vary 1-3 appended characters as part of a hybrid attack. LC4 added the ability to work with prepended characters but the cracking process is much, much slower.

Making the change

Finally on Friday evening, after the end of the regular workday, I implemented passfilt.dll by installing the reference to it into the registry, following the instructions in the Microsoft TechNote referenced earlier in this paper [2]. The registry change is very straightforward. I tested the change by trying to set a too-simple password. I got the appropriate error message. It was working.

In User Manager for Domains I set nearly every password (but not all – see below) to "User must change password at next logon". I also unchecked the “Does not expire” box on almost all of those accounts that had it checked. Finally, under the Policies menu entry, I implemented account lockout to 5 unsuccessful tries. Lockout lasts one hour.

A late-shift worker had an immediate problem with a financial program. Like Exchange, it apparently was using pass-through authentication and balked at the changed credentials. After she logged out and in again and changed her password, the program was fine again.

We had told the employees that *every* password would be changed. Did we actually change *every* password? Not quite. We might have locked ourselves out if we had done so. We left one administrator-level password alone because there had to be at least one known account that could administer the domain. We also left alone the accounts for Microsoft Systems Management Server (many of which are marked “DO NOT MODIFY”), two mission-critical vendor accounts (the vendors would change these on Monday), and a very few generic accounts used to start or run services and where we were unsure of the effects of a change. All other passwords, including those of administrator-level accounts, were changed.

One thing we didn’t change, though, was the password expiration time. It was 120 days and we left it there. I had seen many recommendations that passwords

should expire every 90 days or even 60 or 30, so our time is a little long but we already had complaints about how often passwords had to be changed. This also took into account the human factors. In his article "The Strong Password Dilemma" [7] Richard Smith points out that forcing passwords to be too complex or changed too often will make it more likely that they will be found written down and tucked under the user's mousepad.

Monday morning after the change ...

All network administrators were present and prepared for turmoil on Monday morning. There was very little and it went much more smoothly than the IT team had anticipated. Most employees were able to change their passwords successfully. A small handful triggered the account lockout, which we cleared. Then a few more locked themselves out on Tuesday morning and there have been occasional account lockouts ever since, but generally there was and has been very little fuss.

Results

We performed a baseline audit before the password policy changes and two follow-up password audits in the course of implementation. The results are in the table below:

	# Accounts	% Cracked	# Cracking in 0 seconds	Avg Crack time (hours)
Baseline audit	360	91%	83	2.6
Followup audit	297	64%	14	5.8
Final followup audit	286	57%	0	6.3

The baseline audit

We ran a baseline password audit a couple of days before we implemented the password changes. Using LC3, we imported our passwords from the registry of the PDC and cracked them on a P4, 555mhz machine.

Our cracking algorithm was the moderately strong combination of:

- dictionary attack using the LC3 dictionary of 25,000 words;
- hybrid attack, varying the last two characters of the password, and
- brute force using all combinations of alphanumeric characters.

We ran the LC3 program overnight until it finished. The entire run took 16 hours.

Of the cracked accounts, 83 required zero seconds to crack. Almost all of these passwords incorporated user information but we also had a few users who

preferred excessively simple passwords such as “111111”. If these accounts were excluded, the average time-to-break for the rest was 3.4 hours.

Only 32 accounts survived uncracked: the administrator and (disabled) guest accounts, the generic trainee accounts, and the SMS accounts, which install themselves with strong, random passwords. This wasn't a totally untouched baseline. During my GSEC course I had already changed the administrator, guest, and trainee accounts to harder passwords, in line with what I was learning.

Follow-up audits

I ran the first follow-up audit five days after our password change. At this point it was obvious that we still had some major password security problems, such as the 69 individual but unused accounts with weak passwords. We held a discussion with the manager of the mobile data project and jointly made the decision to delete the unused accounts and move instead to a single generic logon for this project. Considerations in this decision are described below in the discussion of the mobile mapping application. The audit data were not kept.

The next follow-up audit came two weeks after the password change and after the more obvious account problems had been corrected (including deletion of the 69 unused individual accounts). We again used LC3 with the same algorithm as the baseline audit, and the same computer. The cracking run required 20 hours. The results, from the table above, showed that we had improved. The average password crack time was up to 5.8 hours. An average crack time of less than 6 hours did not, however, make us secure.

Public password audits

After two months, we decided it was time to go public with our IT password auditing. Our employees had been informed in the initial email that passwords would be audited but we had downplayed this portion of our program until now. The public audit was preceeded by another educational effort. I sent out an email to all users informing them that there would be a password audit, and that those whose passwords were weak would be required to change them. This email once again included instructions for good passwords.

This time we ran a brief audit designed to catch only the weak passwords. It used only the dictionary and hybrid attacks and required less than 5 minutes. Afterwards employees whose passwords failed were individually notified and asked to change them.

The following day, after everyone had had time to change passwords, we ran the brief audit again. One employee, who had changed his password the first day, was back on the list because the new one was weak too. He was asked to change it again. Amusingly, he replied that he might use as a new password "something you wouldn't want to see". This gave me a chance to let him know that our policy was not to view the cracked passwords but only to test their strength. This is indeed our policy; LC3/LC4 allows the operator to turn off the display of password hashes and cracked passwords and only look at whether a password breaks and how long it took. The SANS course recommended this

course of action. David Sherrod, in a thorough article on password policy [6] in the SANS Reading Room, further explains the possible legal implications of IT knowing passwords. If IT knows the passwords and the company must take disciplinary action against an employee, that employee may successfully argue that he or she was set up.

Final password audit

For this case study, we ran a final strong password audit a few days after the public audits. Our software had been upgraded to LC4 by now but the algorithm and computer were the same as before. The duration of the run was just over 15 hours.

The results in the table show further improvement in overall password strength. If individual users accounts and passwords were the only issues involved, we could sit back now and rest. But the password study uncovered many problem accounts and special situations which have their own security issues and which have caused us many hours of lively debate in IT. These will be discussed in the next section.

Problem accounts and special situations

Passwords are tied to accounts. When we initially examined our lists of accounts during this process we discovered that we had several categories of problem accounts and two sets of special situations that compromised our security.

Problem: Generic accounts

Our baseline audit turned up 360 domain accounts. This number was much more than even the total number of employees. Of these, 66 were generic accounts.

What's wrong with generic accounts? They are mistrusted by many network administrators because there is no individual owner. Because there is no individual owner, a generic account has no real accountability. There is no way to know who has been using the account. We had (and still have) no policy against generic accounts. Some of us in IT had argued against them but we had no consensus and they remain.

Our generic accounts fell into several categories:

- **Built-in accounts**: the Guest and Administrator accounts, which come with NT and can't be removed.
- **Service accounts**: a number of accounts started and ran services. These included the Windows NT Replication user account, the Exchange service account, anti-virus accounts, Microsoft Systems Management Server (SMS) and some others. SMS alone accounted for 20 generic accounts.
- **Template accounts:** these were created for user roles with complex share/permission needs.
- **Vendor/contractor accounts**: some vendors and contractors had generic accounts for interactive logon. Some of these were

administrative-level. Not only were they generic but several were active even though the contractor had long since gone.

- **Autologon accounts:** we had a handful of computers that logged on automatically. Autologon is especially problematic and will be discussed in detail a little later.
- **Student and instructor accounts:** At one time classroom training was done with generic accounts. Later, training switched to using individual user accounts but the generic training accounts remained. They are now used primarily in test and development environments where sample users are needed. They are supposed to be disabled when not in use.
- **Unknown accounts**: We found a small handful of accounts that had apparently been created and forgotten. They were all unused at the time of the audit and most had never been used. Other than clues from the name itself, nobody in the current IT group could remember what they were for.

After the first password audit the IT team looked at all generic accounts to see if any could be eliminated. The mystery accounts were removed. Several others, which might or might not be needed (e.g. an account for backup software) were given a strong password and then disabled. If the software failed, we would know these were actually used.

Problem: Unused accounts

Unused accounts – those which have seen no activity for a period of time – comprised a second category of our problems.

At our baseline audit, we found that we had multiple unused accounts. Of these, 69 were individual accounts which had been created three months previously for a mobile service data application. They were an especially large problem because until this password case study, our practice had been to create initial passwords which matched the username. Not surprisingly, these accounts had the passwords which cracked in zero seconds. Our password practice had worked acceptably well when there was a new employee who was ready and waiting to start using the account, but created a huge risk now that we had accounts sitting idle for months.

Some of the generic accounts had also either never been used, or if used, had not been utilized for a very long time – at least two years in one case. All of the mystery accounts were in this category.

Problem: Automatic login accounts

Finally, we had four generic accounts for machines which were set to login automatically. Autologon is considered another huge security risk. All of these might have strong passwords but autologon makes a password irrelevant. Windows NT autologon works by storing the domain, username and password for the account in the registry – in plain text [1]. There must be a compelling reason to have a workstation logon automatically.

Discussion of the Special situations

Security can be a set of trade-offs between what is risky and what is necessary. Why did we have these special, problem accounts?

Autologon for a 24/7 application

Because we are a utility, our critical systems must be always running. One of the autologon accounts was set up to be sure that certain data would be always available. The application which provided the data was written for a vertical market and did not play by standard rules. It had to be started as a user and could not run by itself as a service. The server was set to reboot if it were to go down, and then the autologon account would start the application. Mitigating factors to the security risk were that the server was physically secured in the locked server room and the account had only user privileges.

Autologon for kiosks

Another three machines were set up to be company intranet/Internet kiosks for a group of employees whose work did not require an individual workstation. These employees also did not have company email or other network access.

As we saw it, our options in this situation were:

1) to give each employee an account and password. We rejected this because: a) their work did not otherwise require an individual computer and b) logging on to an NT workstation takes a non-trivial amount of time. Multiplying this logon time by the number of potential users (60 in the largest office) would turn individual logons into a significant bottleneck. We speculated that human factors would take over and perhaps one employee would share by logging on and letting others use the account.
2) to create a single generic logon for the entire group (about 100 employees in all offices). Every service employee would have to know this username and password. We would probably find it written down.
3) to set the computers to autologon so their intranet and other information services would be immediately available and always up. This is what we did. However these machines are not in a locked room, so making them secure will take more than a strong password. This part of the project is still ongoing.

Mobile service data application

Early this year the company started a project to provide our service data on laptops which could go out into the field with certain employees. These accounts were for the same group of employees for which we had created the autologon kiosks. Our original plan had been to create individual accounts for all these employees. Some of these accounts had been activated but most were not, and this is where the 69 unused accounts of the baseline audit had come from. Our thought processes when we created individual accounts were as follows:

- An individual logon gives accountability, and matches our policy for other employees.

- Service employees rotate duties and might be on call after hours, so we could not predict when any given service employee would use a laptop. The project manager desired that these employees be able to log on without requiring IT intervention. The predictable username/initial password scheme allowed this.

The security risk from the unused logons caused us to rethink our account strategy for mobile data. The project manager still required that the need for IT intervention be kept to a minimum so the laptops could be used after hours and during emergencies.

The working situation for these laptops was:

- The laptops were mounted in locked mounts in service vehicles.
- The vehicles might be left unlocked and unattended for periods of time.
- The particular application contained sensitive data.

Since the application was custom-built for our use, we first suggested that the problem be solved by having the most sensitive items taken out of the application. This was not an option; the data were required. The remaining potential solutions now discussed were:

- Each laptop would have a generic username/password combination that would be different for each individual laptop, but having a predictable pattern. For example, LAPTOP1/password1.
- A single generic username/password would be set for all laptops and known to all users.
- Laptops might be set to autologon.

Setting up a different account for each laptop was a tempting solution but our experience with human factors made us believe this would most likely lead to complete lists of laptop names, usernames and passwords being being written down and passed around among the service employees.

A password-protected screen saver would help protect sensitive data while the laptop was unattended in an unlocked vehicle, but this would require the user to know the password. Autologon and passwords simply do not work together. Also with autologon, an unauthorized person might simply bypass the screensaver by rebooting.

The IT group reluctantly decided to create a single generic username and password for the entire group of laptops – something that could be passed from employee to employee without having to be written down.

This did not initially work all that well. The first password, which I created, was a good strong one that wouldn't crack. It was based on a short passphrase and used punctuation character substitutions throughout. It was so good and so strong that the “mousepad” response mentioned by Smith [7] immediately cropped up and we found the password written down and tucked inside the laptop cases, or stuck under the visors of the service vehicles. So we changed

the password again. The project manager chose the password this time, selecting something he felt would be more comfortable for the service employees but still strong. I sent it out to the supervisors. Since last time I wasn't sure I had made it explicit not to write it down, this time I added that instruction too.

Next Steps

1. ***Secure the autologon kiosks***. This might be a hard one. We may have to stop using autologon because the laptop application, with its sensitive data, is also being installed on these machines and a password-protected screensaver would now be prudent. If we don't use autologon, perhaps we could use the same username/password as for the laptops; it is the same group of employees work who with both. Perhaps we should do something else entirely to authenticate users. One member of the IT team has begun investigating biometric and token methods of authentication.
2. ***Secure SMS***. Minimize the number of generic SMS accounts, and strengthen those passwords that we can assign. Microsoft provides detailed information on SMS security in a white paper [8].
3. ***Implement a formal password policy***. Rewrite the formal password policy to match the lessons learned, then finish the process of getting it approved. The original draft that I wrote no longer matches what we ended up doing. Our experiences were a great teacher.
4. ***Strengthen the administrative passwords***. Use a higher standard for administrative passwords. Audit them regularly with a stronger algorithm than for non-administrative users. They should not break.
5. ***Eliminate null sessions and unsecured shares***. These are fifth and sixth the current SANS Top 20 list. They can make it possible for a non-adminstrative user account to list the system's shares and locate potential network vulnerabilities. How much damage can a non-administrative account cause? With these security vulnerabilities still in the system, and some readily-available probing tools, the answer is "Plenty."
6. ***Keep up with security issues***. The SANS ILOT course demonstrated that ignorance of security issues is no longer an option for IT personnel. The things we don't know might compromise – or might help to secure – our computer systems. This project has made the entire IT team more security-conscious. And if we need more motivation, we can get it from Homeland Security. Our industry was put on Orange Alert twice during the project period.

References

[1] "How to Enable Automatic Logon in Windows NT 3.x and 4.0." Microsoft Knowledge Base. Oct 9, 2002. URL: http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q97/5/97.ASP. (14 Jun 2003).

[2] "How to Enable Strong Password Functionality in Windows NT." Microsoft Knowledge Base. Jun 11, 2002. URL: http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q161/9/90.asp (12 Jun 2003)

[3] Kalis, George J. “Creating passwords that are secure.” Jul 29, 2002. URL: http://asia.cnet.com/itmanager/trends/0,39006409,39068515,00.htm. (14 Jun 2003).

[4] "LC4 – the password auditing and recovery application”. URL: http://www.atstake.com/research/lc/. (12 Jun 2003).

[5] "The SANS Security Policy Project". 2002-2003. URL: http://www.sans.org/resources/policies. (12 Jun 2003).

[6] Sherrod, David H. "Securing Access: Making Passwords a Legitimate Corporate Defense." Jan 15, 2002. URL: http://www.sans.org/rr/paper.php?id=127. (12 Jun 2003).

[7] Smith, Richard. “The Strong Password Dilemma.” Excerpt from Chapter 6 of *Authentication: From Passwords to Public Keys,* Addison-Wesley, 2002. URL: http://www.smat.us/sanity/pwdilemma.html. (14 Jun 2003).

[8] "SMS Security Essentials". Microsoft Systems Management Server White Paper. Mar 22, 2000. URL: http://www.microsoft.com/smserver/techinfo/deployment/20/deploysms/secessentials.asp. (13 Jun 2003).

[9] “Somarsoft Utilities”. Dumpsec version 2.8.2, May 8, 2002. URL: http://www.somarsoft.com/somarsoft_main.htm. (14 Jun 2003).

[10] “The Twenty Most Critical Internet Security Vulnerabilities (Old).” Version 2.504. May 2, 2002. URL: http://www.sans.org/top20/top20_oct01.php. (11 Jun 2003).

[11] "The Twenty Most Critical Internet Security Vulnerabilities (Updated)." Version 3.23. May 29, 2003. URL: http://www.sans.org/top20/. (11 Jun 2003).