



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

SANS Practical

Track 5: Windows Security

Monterey, 2000

Author: Brig Otis

EXECUTIVE SUMMARY

This document is provided in order to meet the practical assignment requirement of SANS Track 5, Fall, 2000.

Securing a Windows NT network is a never-ending job. As soon as you think you have made some serious headway in hardening your systems, yet another exploit makes it way to the headlines of the security newsletters and websites. This paper is not meant to provide you with a means of completely hardening an NT server. There is no “one way” to do so. Depending on the needs of your organization, the various methods presented here may or may not be appropriate for you. While this paper will provide you with a number of ways to harden your system, it is by no means meant to suggest “every” possible way.

There is a “way”, however, that is worth mentioning because it seems that it is a supported method advocated by nearly every security site or authority. The way is “defense in depth”. The concept is that security is not simply a matter of putting up a single barrier or a couple of barriers between you and the outside world. If a bad guy is serious about breaking into your network, you don’t want it to be a cakewalk for them (only provide a few minor hurdles). You want it to be as difficult as possible for them. The way to do this is to present the hacker with hurdle after hurdle after hurdle. With defense in depth, you don’t stop after building and configuring the firewall. You need to look at all the facets of your network. Attention needs to be given to many areas, since attackers will try many different techniques in their attempt at disclosing vulnerabilities. Some of the topics that need to be addressed are presented below. They form the basis of this paper.

- Physical Security / Social Engineering
- User Accounts
- Passwords
- Permissions and Shares
- Services and Ports
- Audit Logs
- Firewalls
- Current Information
- Social Engineering Revisited

As I write this, I can’t help but think about the seemingly futile task of network security. We can go through every checklist out there and read every security newsletter item that crosses our inbox, but as long as the system is connected to the net and powered on, it is vulnerable to exploitation by the bad guys. Throughout this paper, I will be using the term “bad guys” to refer to hackers and others interested in gaining unauthorized access to our systems and the term “net” will refer to either the Internet or an Intranet/Extranet since in either case, we’re talking about a connected computer “network”.

Again, this paper will not discuss every minute detail of securing NT computers, but will hopefully provide the reader with enough information to help the reader in further hardening their system(s).

PHYSICAL SECURITY / SOCIAL ENGINEERING

Threats

- Unauthorized physical access to servers
- Exploitation of 'human' element
- Lack of training for your employees

Defense In Depth

- Create or use a physically secure location
- Educate employees about social engineering and it's effects
- Posted and emailed reminders
- Verification methods (for phone/email)

Discussion

Your Windows NT servers, especially your domain controllers should be located in a secure physical location. A room that is accessible by key-entry only is a good start. Be sure to keep records of who gets copies of the keys that open that room. If there are windows in the walls or doors, be sure that your keyboards and displays face away from them. If your budget allows, install a security camera and keep access to its information (tapes, passwords, etc.) to a minimum.

Many of the exploits available to attackers do not utilize the capabilities of a computer or a computer network. Instead they rely on a process known as social engineering.

Social engineering refers to a non-technological process through which a hacker gains access to a network, typically through the exploitation of a human relationship. Though somewhat preventable through proper training of your staff, it is not possible to harden all the potential avenues social engineering could take.

The normal users represent the greatest threat to your network. After all, they are the ones that you explicitly give access to. Without the proper training, your users might divulge information like usernames, passwords, server or workstation names, or names of software packages running on your network. Any of this information can be helpful to someone trying to gain access to your network. Training users to be careful about divulging this type of information will help you in securing the network, but only to a certain extent. You must also train your domain admins and others with administrative access to not divulge information to the general users. Information to be protected includes the type of monitoring that is performed at your facility. If a hacker learns what is being watched, they can be more careful about being detected.

Recently, Microsoft was the victim of a network intrusion. The following statement was posted at their website shortly after the incident. While they are not specifically addressing social engineering, they could easily be talking about it:

“Many network intrusions originate from an end user's error. These errors happen in organizations of all sizes. It is important for network security managers to assume that such configuration errors will occur and be exploited.”

In the same article, Microsoft advises anyone working in network security to heed the following, one method of minimizing exposure to social engineering exploits:

“Establish a strong password policy and enforce for all users and all accounts. Keep the number of privileged users such as domain administrators on your systems to a minimum, and be sure to audit the activities of such users.”

Kevin Mitnick, the most famous hacker (that got caught) was a star witness in March of 2000 for a panel of U.S. Senators who were considering strengthening federal computer system security. As reported in the Washington Post on March 3, 2000, as a witness, Mitnick stated, “The weakest link in the security chain is the human element”. He said that in more than half of the network exploits he was successful at, he gained information about the network, sometimes including access to the network, through social engineering. In the same article, the Assistant U.S. Attorney who won the conviction against Mitnick stated, “The best security system in the world isn't worth much if you can bypass it by getting security people and other people to give you information--and he was very good at that.”

At a recent SANS conference, an attendee in the Securing Windows NT course related the following story:

He and others were trying to gain access to a hospital's network. They arrived late at night and told the network administrator on duty that they had been sent (by a hardware vendor) to replace the network card in the router. The administrator called his boss to check things out (which made the hackers nervous) and the boss responded with, “Oh yeah...that network card. The password to the router is...”

This illustrates an important aspect of social engineering. All the hacker had to do was gain some confidence with one person and that led to access to fully exploiting the network. The hacker told those of us in attendance at the conference that they had administrative control of that hospital's network for about 3 weeks.

The most important concept to be learned from this section is that social engineering is not something that only affects techies. At-risk people include:

- General Staff
- Receptionists
- Vendors
- Customers

Anyone in one of these groups is susceptible to the effects of social engineering. It only takes one person providing the attacker with enough information to get started and you can soon find yourself falling victim to a sophisticated attack. It is vital that you

educate these people about social engineering and it's possible effects. It is an important step in providing defense in depth.

There are many methods for building your defenses in this area. They include periodic reminders to your staff about the effects of social engineering, posted warnings, and verification methods (like callback) for activities like resetting passwords and providing network information.

USER ACCOUNTS

Threats

- Shared user accounts
- Administrator account
- Guest account
- “null” user sessions
- Untrusted users
- Dormant accounts
- More permissions than required by role

Defense In Depth

- Strong user account policy
- Rename administrator account
- Give Guest account a password and disable it
- Create “honeypot” administrator account
- Disable null user sessions
- Create an “Untrusted Users” user group
- Audit user account activity through event logging and third-party tools like DumpSec

Discussion

A separate user account should be created for anybody who needs to authenticate with the network. Do not use shared user accounts. They do not provide you with an important element in security, “non-repudiation”. Non-repudiation refers to a process in which a person is not able to deny participation in an activity.

All user accounts in your domain can be controlled to a certain extent via the User Manager utility (Start|Programs|Administrative Tools|User Manager). On the Policies menu, choose Account and you will be presented with the following dialog box:

Domain: POORMAN-DOUGLAS

Password Restrictions

Maximum Password Age

☐ Password Never Expires

☒ Expires In Days

Minimum Password Age

☐ Allow Changes Immediately

☒ Allow Changes In Days

Minimum Password Length

☐ Permit Blank Password

☒ At Least Characters

Password Uniqueness

☐ Do Not Keep Password History

☒ Remember Passwords

☐ No account lockout

☒ Account lockout

Lockout after bad logon attempts

Reset count after minutes

Lockout Duration

☒ Forever (until admin unlocks)

☐ Duration minutes

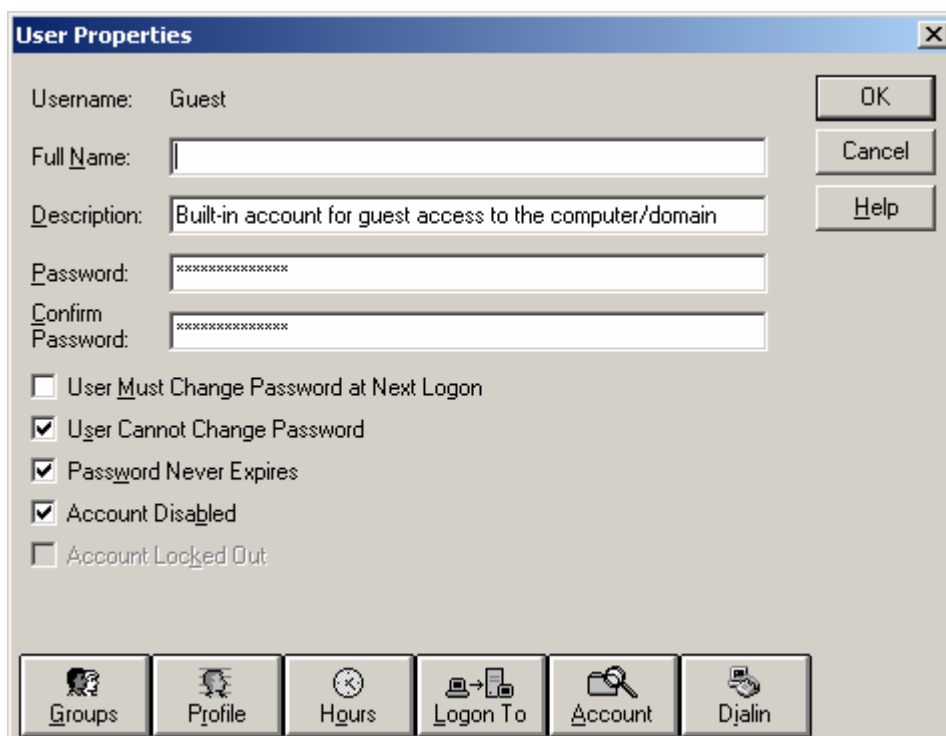
☒ Forcibly disconnect remote users from server when logon hours expire

☐ Users must log on in order to change password

OK Cancel Help

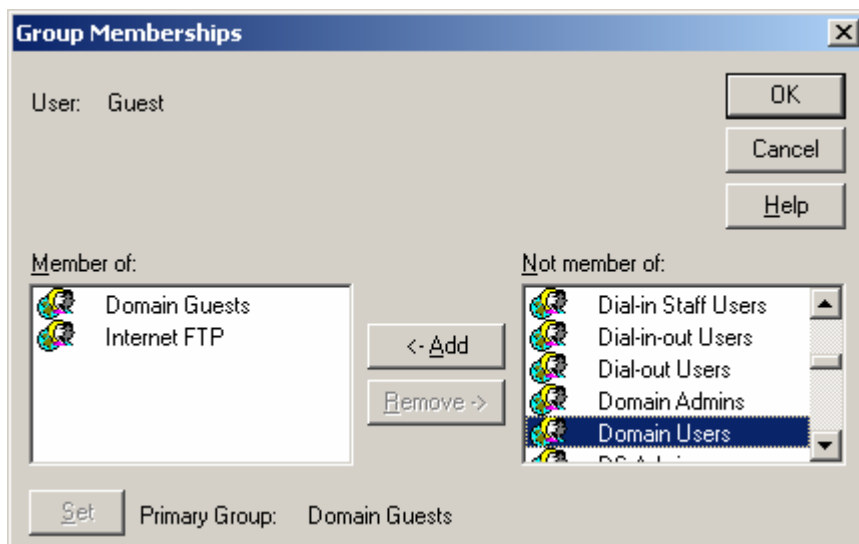
In this dialog box, you can set many of the important attributes of all of the user accounts in the domain. These settings apply to all user accounts. Notice that there is a heavy emphasis on password attributes. The settings displayed above are for a medium-strong environment. Shortening the maximum password age and increasing the minimum password length could enhance NT security.

In the User Manager utility, you can double-click on a user entry to set further attributes to be associated with only that user. In the example below, the Guest account attributes are shown:

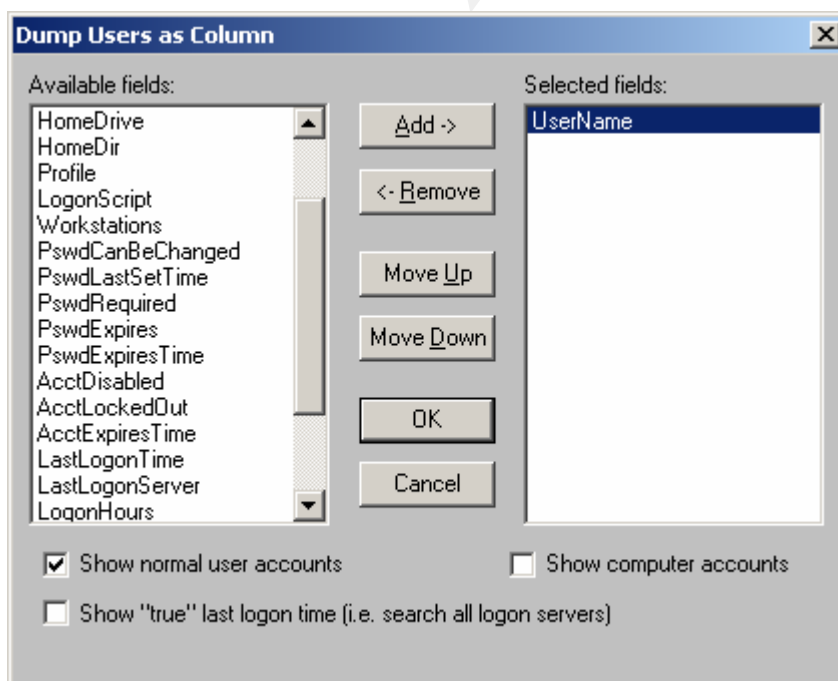


Notice that the account is disabled and the password and confirm password fields have data in them (i.e., the Guest account has a password). The buttons at the bottom allow you to specify this particular user's attributes at a highly detailed level. The options available to you include specifying which user groups the user is a member of, what logon script to use for the user (you may wish to update system files or assign drive letters to system shares via the logon script), the user's home directory, which hours of which days of the week are available for the user to logon (you can specifically set allowed and disallowed hours), which systems the user can log on to (maybe you only want the user to log on to their workstation), account expiration date (optional), and the user account's dialin permission, if allowed.

User accounts should be assigned permissions on a "need-to-know" basis. This means that if a user does not "need" to have permission to a file, directory, process, or system, then they should not be allowed such access. Clicking on the "Groups" button in the User Properties dialog box (shown above), gives you the opportunity to assign which user groups the user is a member of. Group assignments will become more important when you consider the assigning of permissions to files, folders, and services (later in this paper).



User accounts that have been dormant for a period of time should be made inactive. A good rule of thumb on this point is 60 days. There is at least one third-party product that can help you find this information. Somarsoft makes a product called DumpSec (f/k/a DumpAcl) that can provide a number of views on your user account data. In the screen shot below, you can see that you can report on a variety of properties associated with the user accounts on your network, including last logon time. This particular dialog box appears when you choose the “Dump Users as Column” report (there are a number of built-in reports to choose from).



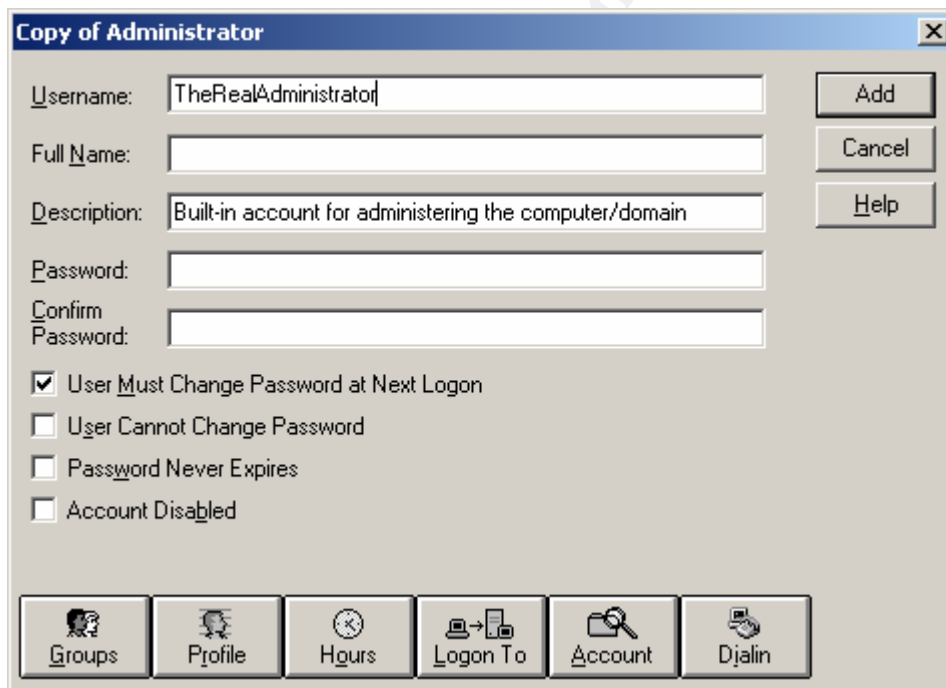
Disable the guest account and make sure the password is not blank. The guest account can not be deleted, but it can be re-enabled by an administrative mistake or an intruder with access to user manager settings (see the User Properties screenshot for the Guest account above).

Prevent “null” user sessions from listing user accounts. Null user sessions represent connections with the network that merely authenticate as a user, but not as a specific user. The authentication is actually based on the null character being used for the username and also for the password. When a null user session is active, it can perform any of the actions allowed for the Everyone group. Disabling null user sessions requires a registry change. Change the value of the following registry entry to 1.

Hive:	HKEY_LOCAL_MACHINE
Key:	\System\CurrentControlSet\Control\Lsa
Name:	RestrictAnonymous
Type:	REG_DWORD
Value:	1

Be aware, however, that some network services (like backups, and virus definition file propagation) may require the use of a null user session to operate.

Rename the administrator account and create a new administrator account with no privileges. In order to do this, you must first copy the Administrator account in the User Manager utility to a new user account with the new user name (see below).



Then, remove all user groups from the Groups property of the Administrator user. This result will be a new account with all Administrator privileges under a new name and an Administrator account that is active, but has no user group memberships (a “honeypot” account).

Audit the “honey pot” administrator account for authentication attempts. Enable auditing in User Manager (choose Policies|Audit; this must be done as a local administrator for the change to take effect) and monitor the event logs for attempts to authenticate by the now defunct Administrator account.

If you have vendors or contractors that periodically need access, create a user group with a name like “Untrusted Users” so that anyone else assigning permissions will be readily able to discern this group from the rest of your users.

PASSWORDS

Threats

- L0phtCrack
- Copy of SAM created via RDISK /S-
- Weak passwords
- Shared passwords

Defense In Depth

- L0phtCrack
- SYSKEY
- Passfilt.dll or other custom password filters
- Strong password policies
- Strong ERD and backup policies and procedures

Discussion

The topic of passwords is one of the most important when it comes to hardening an NT system. Passwords are keys that allow users and systems to access information on our systems or networks.

One of the nastiest programs available for free in the wild is a program called L0phtCrack. It's central purpose is to crack NT passwords. There is a positive side to this program. It can be used to test the strength of your users' passwords. If you are able to determine the weak passwords in your user account database, you can determine what you need to do as an administrator in order to create an environment where password breaking is difficult. If a bad guy obtains the name of a user account on your network, you don't want him or her getting any further. It's all part of the 'defense in depth' concept. L0phtCrack can either be ran live on a network sniffing passwords as they are sent across the network in clear text or it can be ran offline, trying such tactics as brute force to crack the password hashes. Here's how it typically is used.

First, an administrator backs up critical Windows NT operating system files on a server (or workstation) via the RDISK utility. This is part of good backup and recovery preparations. ERD's should be created and updated frequently. They should also be protected.

At this point, it is likely that copies of the SAM database are plentiful, so the next step for a bad guy is to obtain a copy of the SAM database. This can be accomplished via a number of methods:

- The SAM database has a copy stored in the %SystemRoot%\Repair folder which is updated when you run RDISK /S-. An attacker will either run RDISK to cause this backup copy to be created, or they may try to copy an already existing copy. For this reason, you should lock down the permissions on your \Repair folder to only those accounts that need access to it. IMPORTANT NOTE: This is not limited to domain controllers or servers. Any NT workstation or server that has had RDISK /S run on it has a copy of the SAM database in it's \Repair folder.
- ERD disks and tape backups can be stolen. Both of these store copies of the SAM database.
- Social Engineering (discussed earlier in this paper).

Shown below is a listing of the Repair folder after running RDISK /S-. Note the sam._ file which can be copied, uncompressed and imported into L0phtCrack.

```

C:\>dir c:\winnt\repair
Volume in drive C has no label.
Volume Serial Number is CC0E-EB4B

Directory of c:\winnt\repair

11/20/00 12:45p <DIR> .
11/20/00 12:45p <DIR> ..
10/13/96 05:38p 438 autoexec.nt
10/09/98 08:36a 2,510 config.nt
11/20/00 12:45p 29,097 default._
11/20/00 12:45p 14,629 ntuser.da_
11/20/00 12:45p 6,349 sam._
11/20/00 12:45p 14,698 security._
11/20/00 12:45p 2,132,975 software._
11/20/00 12:43p 291,760 system._
10 File(s) 2,492,456 bytes
656,849,408 bytes free

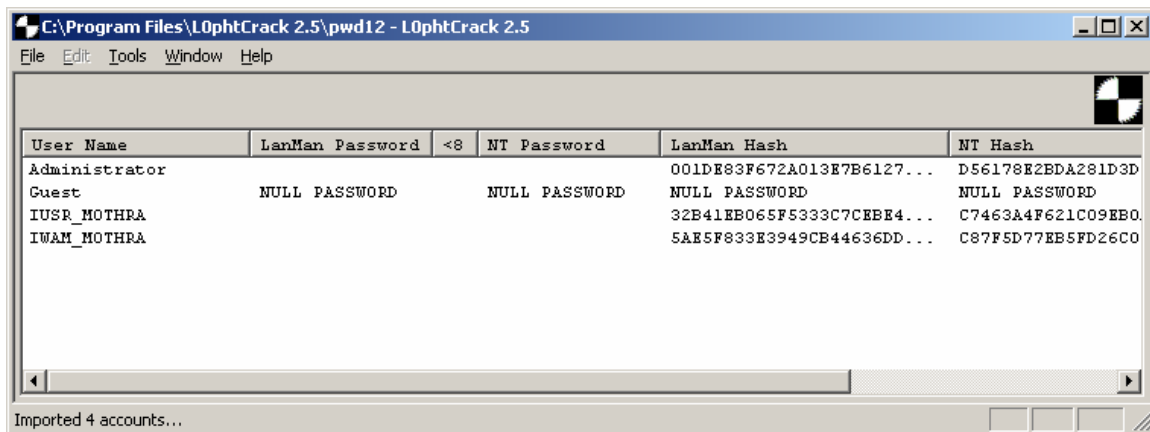
C:\>_

```

Once the SAM database is obtained (or, as in the case of the live network sniffing, some/many NT password hashes are obtained as users log in to the network), the L0phtCrack program can be set to work.

When activated, L0phtCrack will begin to break the password hashes. L0phtCrack can crack in a number of ways, including the usage of custom dictionary word lists. It's salient feature, however, is it's capability to crack any password hash, given enough time, by pursuing a brute force attack, including special characters.

After loading the SAM database copy and instructing L0phtCrack to begin, the usernames included in the cracking will be displayed with properties related to their passwords. Passwords less than 8 characters will be easily identified due to the way that the password hash is created (two hashes; LanMan hash and NT hash). Easy passwords will be cracked in minutes. Medium-strength passwords will be cracked in a matter of hours. Strong passwords will be cracked in a matter of days.



The screenshot shows the L0phtCrack 2.5 application window. The title bar reads 'C:\Program Files\L0phtCrack 2.5\pwd12 - L0phtCrack 2.5'. The menu bar includes 'File', 'Edit', 'Tools', 'Window', and 'Help'. The main window displays a table with the following data:

User Name	LanMan Password	<8	NT Password	LanMan Hash	NT Hash
Administrator				001DE83F672A013E7B6127...	D56178E2BDA281D3D
Guest	NULL PASSWORD		NULL PASSWORD	NULL PASSWORD	NULL PASSWORD
IUSR_MOTHR				32B41EB065F5333C7CEBE4...	C7463A4F621C09EB0
IWAM_MOTHR				5AE5F833E3949CB44636DD...	C87F5D77EB5FD26C0

At the bottom of the window, it says 'Imported 4 accounts...'.

L0phtCrack has an interface that is easy to work with (shown above). Remember that you can use this tool, just as well as the bad guys can. The difference is that you can use it to perform password strength analyses on your own domain's user accounts. For instance, in the example above, you can see that the Guest account on the target machine does not have a password assigned to it.

This brings us back to the concept of defense in depth. You can assume that someone will try to crack a copy of your SAM database with either L0phtCrack or a similar program. With that in mind, you should now be thinking about the following:

- Strong password policies
- ERD policies and procedures
- Backup policies and procedures

A strong password policy will require the following of passwords. They should:

- be at least 14 characters in length
- be composed of mixed case characters, numbers, and non-standard characters
- not be shared with others
- not be easily guessed
- be rotated faster than L0phtcrack can crack them
- have minimum and maximum ages
- not be used again (password history)

Although difficult to require via a password policy, a stronger measure is to include extended ASCII characters in your passwords. A custom password filter could be written for this purpose or you can implement passfilt.dll. passfilt.dll is a password filter provided in Service Pack 3 and later. When enabled, it will require passwords to be 6 or more characters long, not include the user's name in it, and include mixed case characters, numerical characters, and non alphanumeric characters/symbols. It is enabled by adding the following value to the registry:

Hive: HKEY_LOCAL_MACHINE
Key: \System\CurrentControlSet\Control\Lsa
Name: Notification Packages
Type: REG_MULTI_SZ
Value: PASSFILT

Strong ERD and backup policies and procedures should include considerations for physical security, accountability and non-repudiation by administrative staff members.

There is a utility shipped with Service Pack 3 or later that can be used to encrypt the SAM called SYSKEY.EXE. SYSKEY encrypts the passwords in the SAM database and generates a 128-bit key that you use as an administrator to start the NT operating system. Once implemented, NT will not start without this key. The key is provided to NT via three methods. It can be stored on a diskette, stored on the system itself, or created from a password. You can choose the method you want to use when you implement the SYSKEY utility. If you choose diskette or password, the diskette or password will need to be available at bootup or the system will not boot. If you choose the system method, the downside is that the key is stored on the system itself.

To implement SYSKEY, from a command prompt, enter “syskey” and press Enter. You will then be prompted for the following:



If you select the “Encryption Enabled” option, you will be presented with a confirmation box reminding you that, “Once encryption is enabled, it can not be disabled. Make sure you have a current Emergency Repair Disk before continuing”. Clicking OK to the confirmation box will provide you with a choice of the implementation options described above (diskette, password, or system key).

SYSKEY protects the local SAM, so it must be implemented on all systems in order to provide domain-wide protection. The implementation method you choose does not need to be the same on each system, but it is recommended that you use the same method across all systems for the sake of consistency in administration. It is also recommended that you create an updated ERD prior to running SYSKEY (as you should prior to any instance of a system change).

While implementing SYSKEY will certainly enhance your defenses against attack, it should be mentioned that there is a vulnerability in SYSKEY when applied to

systems prior to Service Pack 6. A patch was released in late 1999 for previous system versions. The vulnerability has to do with some instructions in the SYSKEY implementation that repeat initialization data when encrypting using SYSKEY, providing cryptanalytic devices a means for “reading” the crypto keystream (re-using the keystream is not a recommended practice for encryption).

SYSKEY does not prevent utilities like L0phtCrack from sniffing passwords traversing the network. So, L0phtCrack and utilities like it still seem to be our greatest threat at this point. Never fear, however, since there is an authentication scheme available with Service Pack 4 or later that will prevent L0phtCrack from sniffing passwords. It is called NTLMv2.

Because of the way that password hashes are computed and stored, utilities like L0phtCrack can use precomputed dictionaries of password hashes to find matches with those found in a SAM database. Since all of the password hashes prior to NTLMv2 are computed in the same way, it is not terribly difficult to work backwards through the process of creating a NTLMv1 password hash.

NTLMv2’s salient features include mutual authentication between client and server and the use of a “salt” (random input) when encrypting the hash. These features make man-in-the-middle attacks useless and L0phtCrack is unable to crack the password hashes.

There is a registry value that can be modified to tell the system whether or not to use NTLMv2 authentication.

Hive:	HKEY_LOCAL_MACHINE
Key:	\System\CurrentControlSet\Control\Lsa
Name:	LMCompatibilityLevel
Type:	REG_DWORD
Value:	1

It’s value can be set to any value between 0 and 5 inclusive. The different levels that it can be set at determine what type of authentication the system will use in which circumstances. For instance, setting the value to a level of 1 (the recommended value) will tell clients to attempt to negotiate NTLMv2, but fall back to NTLMv1 when necessary. Setting this value on a domain controller will tell it to accept NTLMv2 authentication if the client requests it.

PERMISSIONS/SHARES

Threats

- Unauthorized access to critical data
- Too much authority given to low-level users
- Default “Everyone” setting

Defense In Depth

- Assign permissions to user groups instead of individual users
- Regularly review permissions, especially critical servers / data
- Assign permissions on a “Need to know” basis
- Educate any users with “Full Access” on the difference and interaction between NTFS permissions and share permissions

Discussion

There are two types of permissions that can be set, “share” permissions and “NTFS” permissions. Although they are exclusive of each other, they can produce unexpected results when applied together if they are not understood and used correctly.

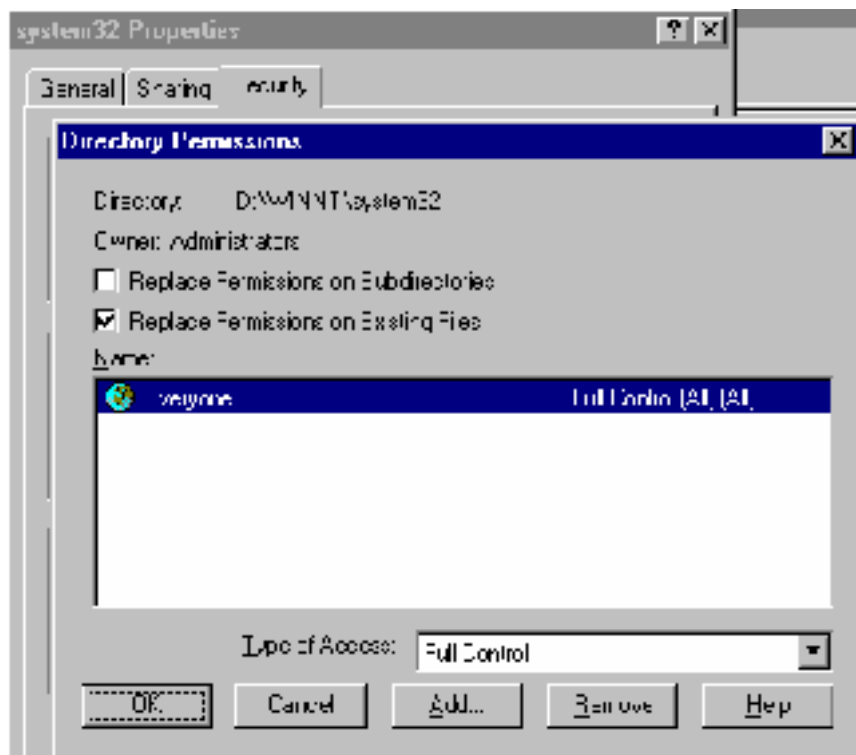
First, you should understand what NTFS permissions are. NTFS permissions are the last checkpoint prior to an authenticated user gaining access to a file. The only exception to this is if you don’t disable null user sessions since a null user is a member of the Everyone group and does not technically “authenticate” with the network (see the section on user accounts). NTFS permissions can be assigned to folders or individually to the files inside folders. When assigning NTFS permissions to a folder, the default is to assign them to all of the files and subfolders within the folder, although this option can be unchecked when assigning the permissions. Another option available is to apply the permissions to all of the subfolders and the files within them as well. Care should be taken when doing this though, as you can easily overwrite permissions further down the folder nesting chain without realizing it.

Generally, the permissions you assign should be for user groups instead of user accounts, since it’s easier to add and remove users from groups than it is to find all of the places that a user might need to have permissions removed or assigned. By default, all files and folders allow the Everyone group to have full control.

It is important to review the files and folders on your NT Server for appropriate permissions. The Somarsoft tool, DumpSec can be used for this purpose. Earlier in this paper, you saw how it could be used to review user account information like last login time. You can also use it to show you permission information on the server. For instance, you can instruct DumpSec to provide you with a list of all files and folders that have different permission settings than a given parent folder. Using the tool in this way allows you to narrow your search for vulnerabilities on the server.

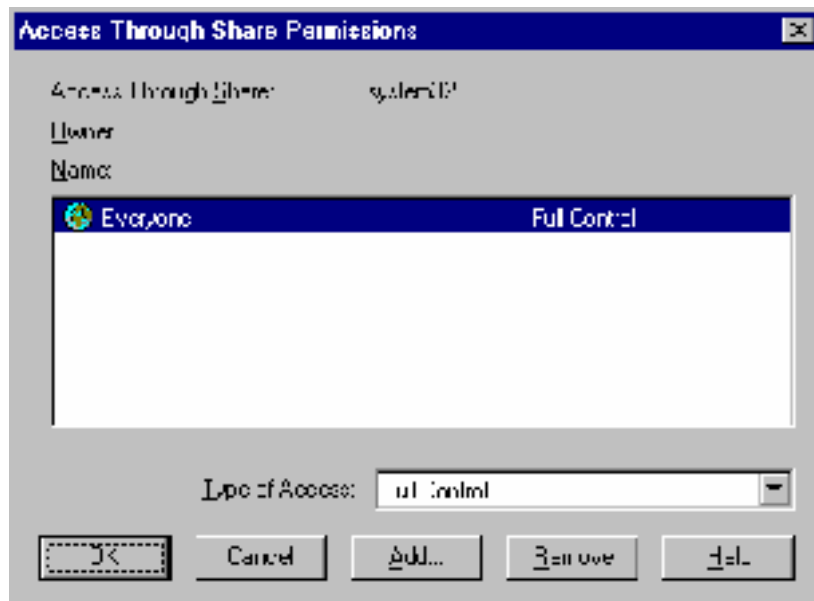
Usually, the highest level of NTFS permissions you should give any user account or group is change. The only difference between change and full control is that full control allows you to assign permissions. If you have a staff member that is responsible for assigning permissions, they will need to have full control.

The screenshot below shows the default permissions for the system32 folder. Note that anyone could replace system files with Trojan horse versions of their own, given these permissions.



Share permissions apply to shared folders. They can not be applied to files. When you share a folder with others, the default setting is to allow the Everyone group to have full control. This can be viewed as allowable if you have set the NTFS permissions in a way that will restrict the various classes of user.

The screenshot below shows the default share permissions for the system32 folder. Note again the highly non-restrictive default setting.



Share permissions and NTFS permissions interact in such a way that the most restrictive setting between the two types of permissions for a given user will be the most access allowed to that user.

As an example, consider a folder shared as “Ashare4You”. You are logged into the network as user “joeuser”. The share permissions are “Everyone – Full Control” and the NTFS permissions on the folder (and the files within it) are “Domain Users – Read (only)”. Through the share you have full control permissions, but when you attempt to interact with one of the files, you will find that you only have the Read permission. If the share and NTFS permissions were switched, you would have Full Control over the folder and the files within it if you were logged in at that computer. However, if you were accessing the shared folder over the network (through the share), you would only have Read permissions.

While it is convenient to be able to share folders across the network, it is very important to have the tightest controls on them as possible, since sharing them makes them accessible by outside parties. Typically, a hacker will scan for shares on a target host to see if there is a way that he/she can exploit them. If a folder with “Everyone – Full Control” (the default setting) is made accessible to a hacker and you have not disabled null user sessions, the hacker can gain full control over the folder and the files within.

REMOVE NON-ESSENTIAL SERVICES & BLOCK NON-ESSENTIAL PORTS

Threats

- Services and open ports can provide network topology
- Simple commands can expose the services and ports
- Denial-of-service attacks

Defense In Depth

- Use netstat and nbtstat to find your openings
- Port filtering

Discussion

Services that are running can provide information about your network, the workstation/server that they are running on, or the user logged on at the workstation/server. As such, they can be used as intrusion vectors by the bad guys. Removing non-essential services helps to further defend your network against attackers.

When a bad guy initiates an attack on your network, the worst thing you want to do is provide them with more information than absolutely necessary. Again, defense in depth is what you're after and you do not want to leave services running that can be accessed to acquire information about your network or leave ports open that are not being used by any necessary services or features.

The first thing a bad guy is going to do is attempt to gain some information about the topology of your network. This may begin with a simple scan of an IP address range, looking for any responses indicating the existence of a device at the address. Next, the attacker will try to find out just what the device at the address will listen to.

To find out which ports are available on your server, you can run the **netstat** utility from a command prompt. When you run **netstat**, you can find out which ports are open, what protocols are running for those ports, and what address or address range the ports apply to.

The example below is displaying all connections and listening ports (the **-a** switch) and the addresses and port numbers are being displayed in numerical form (the **-n** switch).

```

D:\WINNT\System32\cmd.exe
D:\>
D:\>netstat -a -n

Active Connections

Proto Local Address          Foreign Address         State
TCP    0.0.0.0:135             0.0.0.0:0               LISTENING
TCP    0.0.0.0:136             0.0.0.0:0               LISTENING
TCP    0.0.0.0:139             0.0.0.0:0               LISTENING
TCP    0.0.0.0:1400            0.0.0.0:0               LISTENING
TCP    0.0.0.0:14776           0.0.0.0:0               LISTENING
TCP    0.0.0.0:30000           0.0.0.0:0               LISTENING
TCP    0.0.0.0:36952           0.0.0.0:0               LISTENING
TCP    0.0.0.0:39371           0.0.0.0:0               LISTENING
TCP    123.123.123.123:137     0.0.0.0:0               LISTENING
TCP    123.123.123.123:138     0.0.0.0:0               LISTENING
TCP    123.123.123.123:139     0.0.0.0:0               LISTENING
TCP    127.0.0.1:1020          0.0.0.0:0               LISTENING
TCP    127.0.0.1:1020          127.0.0.1:1020          ESTABLISHED
TCP    127.0.0.1:1020          127.0.0.1:1020          ESTABLISHED
TCP    127.0.0.1:1020          0.0.0.0:0               LISTENING
UDP    0.0.0.0:135             *:*                      *
UDP    123.123.123.123:137     *:*                      *
UDP    123.123.123.123:138     *:*                      *
D:\>

```

Take note of the UDP 137 and 138, and TCP 139 ports that are open. These are “signature” ports that Microsoft uses. Seeing these ports displayed here tells you that this is a Windows operating system.

There is another command-line utility called **nbtstat** that can, as described in Windows Help, “be used to display protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP)”. It can reveal some very interesting information about your system.

```

D:\WINNT\System32\cmd.exe
D:\>nbtstat -A 123.123.123.123
NetBIOS Remote Machine Name Table

Name                Type               Status
-----
BRIGLAPG1           <00>               UNIQUE             Registered
BRIGLAPG1           <20>               UNIQUE             Registered
BRIGDOMAIN           <00>               GROUP              Registered
BRIGDOMAIN           <1C>               GROUP              Registered
BRIGDOMAIN           <1B>               UNIQUE             Registered
BRIGDOMAIN           <1E>               GROUP              Registered
BRIGDOMAIN           <1D>               UNIQUE             Registered
__MSBROUSE__         <01>               GROUP              Registered
BRIGLAPG1           <03>               UNIQUE             Registered
BOTI                 <03>               UNIQUE             Registered

MAC Address = 20-4C-4F-4F-50-20

D:\>

```

In the example above, we can readily determine a number of things. In order, the entries from the list in the screenshot above tell us the following:

- The workstation service is running
- The file server service is running
- The domain name is “BRIGDOMAIN”
- The server at this address (123.123.123.123) is a domain controller
- It is the domain master browser
- It will accept browser service elections
- It is a master browser
- It is a master browser
- The messenger service is running for this server (named “BRIGLAP01”)
- The messenger service is running for user “BOTI”
- The MAC Address is 20-4c-4f-50-20

This is quite a bit of information gained by entering a simple NT command-line command directed at a particular address (123.123.123.123). The information gained from this activity can be extremely useful to an attacker. For instance, now that the attacker knows that this is a domain controller, it might be useful for the bad guy to cause the server to run the RDISK /S- command (which will create a copy of the SAM database in the %SYSTEM%/Repair directory), copy the SAM database to an offline location, and go after it with L0phtCrack or some other password cracking utility. Further, once the attacker gets the SAM loaded into L0phtCrack, it is very likely that they will be most interested in the password for the BOTI user. This is likely to be true, since it is very probable that the user logged in at the time this command was run (BOTI) is an administrator.

Every network has it's own particular needs. There is no way to state exactly which services are needed and which are not. There are many references available to determine the setup that is best for you. For example, in the case of an IIS server (web server), the following services are the only services required to use IIS (remove the others):

- Event Log
- License Logging Service
- Windows NTLM Security Support Provider
- Remote Procedure Call (RPC) Service
- Windows NT Server or Windows NT Workstation
- IIS Admin Service
- MSDTC
- World Wide Web Publishing Service
- Protected Storage

LOGS

Threats

- Unauthorized access to data
- Undiscovered vulnerabilities

Defense In Depth

- Enable auditing
- Sweep logs to secure off-server location
- Deploy log analysis software

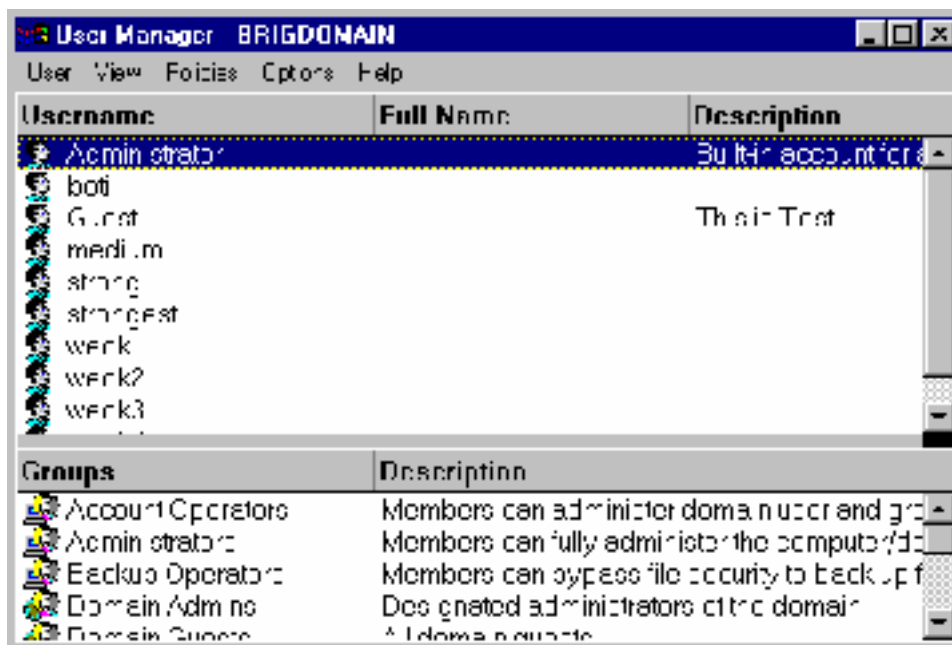
Discussion

You should enable logging on your servers and workstations. This is especially true for critical files, directories, and services. A good place to start is your ...\\winnt\\system32 directory. By default, this directory contains most of the critical files in your Windows NT system. In an ideal situation, all of the following aspects of logging are in place:

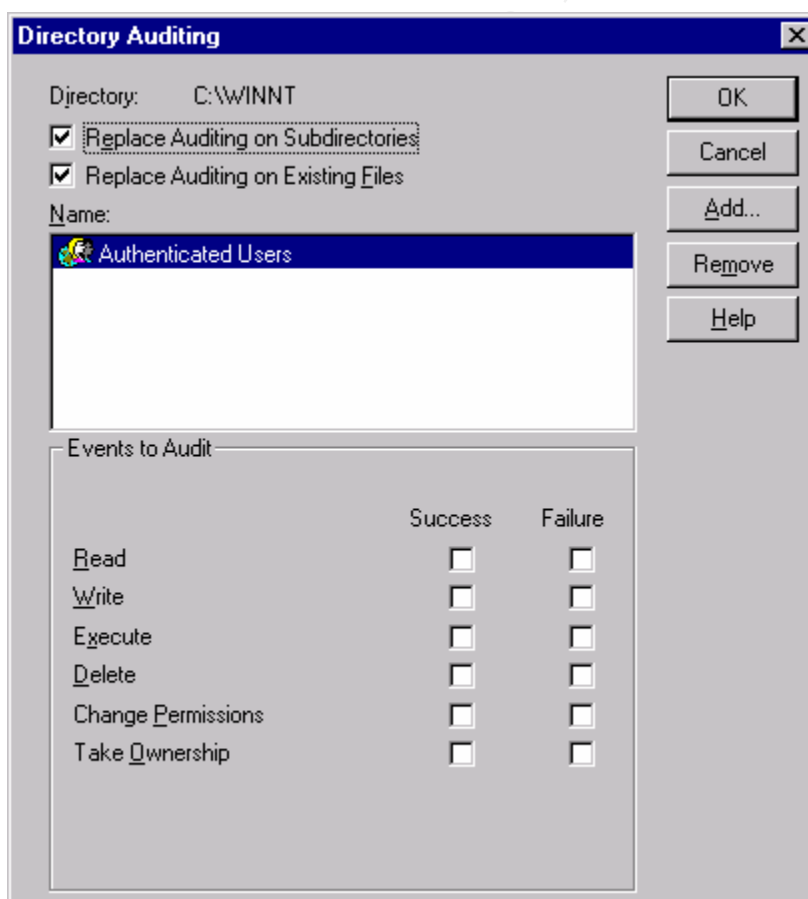
- Clocks on all systems are synchronized. This enables a more accurate audit trail across systems.
- Log files are stored in a secure location. Offsite storage for archival purposes. Secure location off of the production servers, if possible.
- Log files are stored on a different server than the one they are being generated on. This enables efficient analysis of log traffic and makes it more difficult for an intruder to cover their tracks by deleting or modifying log files.
- Log analysis software is constantly running, looking for anomalies and other events that you have identified as necessitating alerts. Train the software on what is “normal” for your environment and be sure you understand the alert messages and what they signify.

It is important to keep the above recommendations in mind, as well as it is important to set the standard NT logs correctly to provide maximum protection (through auditability).

There are three standard event logs in NT. They are the System, Security, and Application logs. The system and application logs will normally start populating with events as soon as you install the NT operating system. The security log will start populating after you login as a local administrator, and turn auditing on in User Manager (choose Audit on the Policies menu in User Manager).

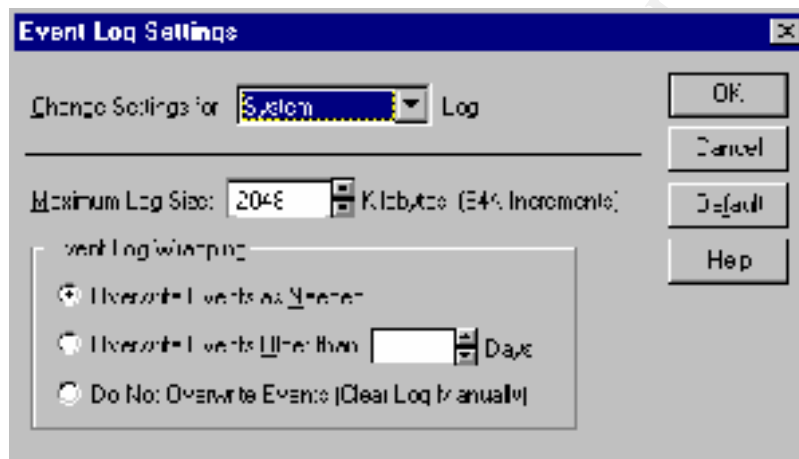


You can instruct NT to audit in more detail via the Security Tab of the Properties dialog for files and folders.



For each of the three event logs, you can instruct NT how you want the log configured. In the display below, the settings for a System Log are shown. Note that the “Overwrite Events as Needed” option is selected. The benefit from this selection is that you will not fill up your hard drive with logged events (nor will an attacker be able to flood the log to the extent that they would cause a DOS scenario by jamming your hard drive with event log entries. The disadvantage of this setting is that an attacker can flood the event log after carrying out their activities in order to push the events signifying their actions past the log wrapping limit (push the events off the back end). The other two settings do allow you to audit events over a broader time span, but there is no safety measure built in to these options to prevent your hard drive from filling up.

One scenario you may want to consider is instructing the system to not overwrite events and have a system in place that takes the events from the event log and writes them to a secure off-system location for further analysis. Of course, you should also have the system set to alert you if the hard drive begins to fill up. In this way, you will have more information available to you for further analysis and auditing.



FIREWALLS

Threats

- Multitude of attackers and available attacking
- Undiscovered vulnerabilities

Defense In Depth

- Deploy a network firewall
- Use network address translation on the firewall
- Deploy monitoring software

Discussion

Deploy a firewall between your internal network and the internet. Make all internet connections take place through this firewall. You should regularly review the policy settings on your firewall to make sure that it only allows necessary traffic through its ports. Common ports that are open include tcp ports 21 (ftp), 25 (mail), 80 (http), and 443 (ssl). Typically, these ports are configured to allow connections that are initiated from within your network only, although you may have reasons to allow connections that are initiated from outside your network. For example, you may be hosting a website that needs to be available to the outside world. In this case, you would allow port 80 connections to be initiated by external parties, but only to the IP address of the website.

Firewall solutions are composed of both software and hardware. Network Address Translation (NAT) servers, switches, routers, and the software that run on them can provide you with all the tools you need for filtering and analyzing protocol traffic. Filtering protocol traffic is necessary if you wish to protect your company's investment in its computer network and the productivity it can afford. Without protocol filtering provided by a firewall solution, your network will soon be compromised.

Due to the nature of the internet, we must always be on the lookout for the enemy. The 'enemy' can be any person or service connected to the internet. This is not to say that everyone and every service is bad, it simply is stating that there are bad people out there and by taking the proper steps in securing our networks, we become less likely to be a 'target' of theirs. And, they are always looking for targets.

A hacker usually begins their activities by surveying the internet for likely targets. This is accomplished by a number of means. They may:

- visit chat rooms where other hackers are describing targets
- read the business section of the local paper looking for a startup company that may not have all of their defenses up yet
- visit a newsgroup in which a network administrator is describing some technical difficulty they are facing
- use the NSLOOKUP.EXE utility to determine IP addresses and host names

- initiate a port scan on your exposed IP addresses to find out which ports are listening

CURRENT INFORMATION

Threats

- Rapidly changing technical environment
- Undiscovered vulnerabilities
- Speed of information dissemination (to the bad guys)

Defense In Depth

- Subscribe to security newsletters
- Apply latest service pack
- Visit security sites
- Join hacker chats (but don't reveal any of your network's info)

Discussion

Staying current with the latest news regarding Windows NT security is one of the most important steps you can take. You can bet on the bad guys watching the discussion groups and the security headlines for new, un-patched exploits.

At the Monterey conference, there was an exhibition set up called ID Net. It consisted of a "hardened" target host and an open challenge to crackers to try and break in. From Sunday to Wednesday, the host remained untouched. On Tuesday, Microsoft released a security bulletin concerning IIS 4.0 and 5.0 (see Microsoft Bulletin MS00-078 at <http://microsoft.com/technet/security/bulletin/fq00-078.asp>). As you may have guessed by now, the champion cracker eventually defeated the host using the exploit revealed in the Microsoft Bulletin. That bulletin was only two days old when the cracker got through. The point to be learned from this is that if you don't stay up to date with the latest Microsoft patches and Service Packs, you could be leaving yourself wide open to similar exploitation.

There are many ways to keep yourself up to date. For starters, your NT computers should all be at service pack 6a. You can determine what service pack level you are at by typing "winver" at a command prompt. To arrive at a command prompt, click Start | Run | type "cmd" & press Enter. By ensuring your service pack level, you are confirming that the latest known vulnerabilities now have their fixes loaded on your system. One thing to remember is that if you install or reinstall components from the NT Option Pack or an I386 directory, it is possible that you will end up downgrading certain system files to pre-service pack versions. One way to ensure that you are always installing the correct version, is to replace the files in your I386 directory with the corresponding uncompressed files from the latest service pack.

You should also browse to <http://www.microsoft.com/security> and start reading. There are many links on this page leading to security topics specifically related to just about any Microsoft NT security question you have. While you are there, sign up to receive Microsoft's security bulletins via email.

You should also keep your staff up to date with current information. As the NT networking environment continues to develop, so should your people. Educate your people at least three or four times a year on the effects of social engineering. Provide adequate training for anyone participating in security-related activities as a part of their job.

It also pays to have current information about your own network and servers. ERD's should be ran frequently and the disks should be available to top-level administrators in case of issues (note the security risks concerning the copy of the SAM database outlined above). Current drive images can help keep you ready to recover as well. For instance, if you are hosting an IIS site, consider hosting the content in a secure location and instructing IIS to pull the content from that location for delivery. If your IIS server crashes, you can swap drives and continue on your merry way.

Adequate backups are also a consideration for keeping current information. Backups should be performed at least daily on all critical information. The backup media should be rotated offsite for later use, if necessary.

SOCIAL ENGINEERING REVISITED

I can not emphasize enough the importance of considering the impact that social engineering can have on your organization. A figure given at the SANS conference was that over 90% of the financial losses suffered by organizations via system attacks was due to errors and attacks by it's own people. This tells us that one of the areas we can implement the greatest good in is by instructing our people how to respond to social engineering ploys (whether they are in fact real attacks or not).

Many of the posted practicals at the SANS website miss this point entirely (or gloss over it so as to provide a more 'technical' paper. As part of this paper, we are supposed to incorporate and expand on previously posted papers. This implies that I must find deficiencies in others' work. So, apologies aside and a social engineering theme to carry me through, here is what I found in a couple of the posted practicals.

In Alex Park's posted practical, Alex tells us that "there are three main techniques for acquiring passwords: manual, automated, and (network) sniffing". While these methods are indeed used, the method that is missing is that of social engineering. There are many examples available on the Internet of social engineering and reverse social engineering events where users have given their passwords to unknown individuals. It is vital that everybody with an account on your network be instructed to not give their password to anyone, even to the system administrators (sysadmins can reset passwords; they don't need to know the old one; non-sysadmins can not reset passwords and they do not need to know others' passwords). One method of implementing this safety consciousness in your organization is to have all users sign a network usage agreement

that specifically instructs them to not share their password and outlines various remedies available to the company should users choose to not follow that directive.

Don Michelli's document which outlines eleven best practices for securing Windows NT covers a laundry list of actions (mostly registry hacks) that can be taken in hardening an NT system. Similar lists are available at a number of sites on the Internet:

- <http://microsoft.com/technet/security/tools.asp>
- <http://afcert.csap.af.mil/winntcheck.html>
- http://www.rcmp-grc.gc.ca/tsb/pubs/bulletins/bull45_3.htm
- <http://www.win2000mag.com/Articles/Index.cfm?ArticleID=3571>
- (others)

An interesting theme running through these lists (as well as the Michelli document) is the absence of any references to social engineering. Again, the technical side is important, but you can not ignore the effects that the human element can play on your system.

In fact, if I were to offer a recommendation for securing Windows NT in a sentence or two it would be to follow the checklists available, making sure to customize the decisions for your own environment, secure the SAM database, and educate yourself and your users on the effects of social engineering.

© SANS Institute 2000 - 2002, Author retains full rights.

REFERENCES

- Beyond-Security's SecuriTeam.com, **Patch Available for the Syskey Keystream Reuse Vulnerability**,
http://www.securiteam.com/windowsntfocus/Patch_Available_for_the_Syskey_Keystream_Reuse_Vulnerability.html, December 16, 1999.
- Carnegie Mellon Software Engineering Institute, **Using SYSKEY to protect the password data for Windows NT 4.0**, <http://www.cert.org/security-improvement/implementations/i028.02.html>, March 17, 1999.
- Computer Security Institute, **Social engineering: examples and countermeasures from the real-world**, <http://www.gocsi.com/soceng.htm>, Reprinted from the November, 1999 issue of the Computer Security Alert, CSI's monthly newsletter.
- Fossen, Jason, and Kolde, Jennifer, **Securing Windows NT, Step-by-Step**, The SANS Institute GIAC Training, October 2000.
- Lemos, Robert, **Mitnick teaches 'social engineering'**, ZDNet News, <http://www.zdnet.com/zdnn/stories/news/0,4586,2604480,00.html>, July 17, 2000.
- Microsoft Web Site, **Information About Security Incident on Microsoft Corporate Network**, <http://www.microsoft.com/technet/security/001027.asp>, November 1, 2000.
- Microsoft Web Site, **Windows NT 4.0 Member Server Configuration Checklist**, <http://microsoft.com/technet/security/mbrsrvcl.asp>, June 6, 2000.
- Schwartz, John, Washington Post, **Hacker Gives a Hill How-To Mitnick Tells Panel Loose Lips Sink Systems**, <http://www.washingtonpost.com/wp-srv/WPlate/2000-03/03/1831-030300-idx.html>, March 3, 2000.