Top 10 Mistakes on Windows Internal Networks

Deirdre Hurley

Name: Deirdre Hurley

Assignment: GSEC V1.4b Option 1

Title: Top 10 Mistakes on Windows Internal Networks

Table of Contents

Abstract

In this paper I aim to highlight ten common mistakes on Windows systems, which make the job of a disgruntled employee or a malicious attacker who manages to get past your firewall, far easier. All of the mistakes are in relation to Microsoft Windows operating systems, as my past experience at conducting internal network assessments, has shown me that the easiest way for an attacker to get onto any internal network is via these high-risk Microsoft Windows vulnerabilities. As well as this, because backward-compatibility is a feature of Windows systems, all of these mistakes apply to both Windows NT and Windows 2000, after all Windows 2000 is based on NT technology. For each mistake outlined, a tool or a technique, which will aid the system administrator in identifying if the problem exists on his/her network, is recommended. Using these command line tools is important for every system administrator, as it is the best way to actually get to know the systems on your network and to realise that these are the actual tools that any hacker will use. Finally, solutions to mitigate the risks presented are also discussed.

Introduction

The threat to networks continues to grow due to the development of new attack techniques by hackers. Software attack tools are readily available on the Internet and because many tools now feature simplified graphical user interfaces (GUIs), unskilled or novice hackers are joining in. This development opens "hacking" to a much wider cross-section of the computer-using population, including an organization's own non-technical employees.

Paul Desmond refutes the idea that more attacks originate on an internal network than from an external source [1], while Terry Boston states that approximately "80% of attacks are from within the organisation" [1]. Although it is clear that not everyone is in agreement on the exact percentage of attacks that are generated on the inside, there is general consensus that external "hackers" are not the only source of attacks and data compromise. Furthermore, according to SANS GSEC "insider attacks have a higher rate of success because they are carried out by people with inside knowledge about (and often some level of existing access to) your systems, networks and data".[2]

The following 10 mistakes, which are found on many internal Windows networks, serve to aid an attacker. Systems administrators who rectify these mistakes and who ensure that this process is done on an on-going basis, will not only improve security but will also gain knowledge of the characteristics of one's network or environment. Finally, while reading this paper keep in mind the following words of wisdom provided by SANS GSEC – "You can't plan a defence without knowing the offence". [3]

[1] Boston, Terry "The Insider Threat" [26]
[2] SANS GSEC Section 5.6 "Windows Auditing" p10
[3] SANS GSEC Section 5.7 "IIS Security" p24

1. Allowing Null Sessions

This vulnerability is also known as the "Red Button" vulnerability and in order for a Windows system administrator to realise it's importance to a malicious user, a quote from McClure is required – "[null sessions] can be the single most devastating network foothold sought by intruders". [4]

A null session is a session established with a server when no credentials are supplied. Microsoft Windows NT and Windows 2000 by default allow this type of connection. This anonymous connection is used by applications to list account names and enumerate shares on remote servers. Examples of these applications include the Windows NT ACL and Windows NT Explorer, which require anonymous connections to list account names and enumerate shares respectively [24]. (Joe Finamore also provides a more in depth discussion on why exactly Microsoft when to the trouble of supporting null sessions. [17])

The syntax for creating an anonymous connection is:

net use \\ipaddress\ipc$ "" /user:""

This command connects to the hidden interprocess communications share (IPC$) at the specified IP address, as the built-in anonymous user (/user:"") with a null ("") password.

However, as Timothy Mullen outlines in his paper, "a user with no credentials, can be used to glean a tremendous amount of information from your network without raising any eyebrows" [5]. This is possible because the Null session has the same permissions as the built-in group Everyone, meaning on the default install the Null session has remote access to many areas of the registry. Consequently, a totally anonymous user has the ability to connect to the IPC$ share of a server and run the registry editor (REGEDT32.EXE) to view, and even change some registry keys. Moreover, this anonymous connection can also be used by users to download all usernames, groups, administrators, password change dates, last login dates, account policy, trust relationships, lockout policy, etc with standard utilities which are provided by Microsoft. This information can dramatically increase the chances of a user successfully guessing ID and password combinations to gain access to Windows Domains.

In order to prevent users from enumerating the type of information outlined above the "Restrict Anonymous" key should be enabled in the registry for Windows NT, while Windows 2000 provides a graphical interface via the Security Policies MMC snap-in. The SANS GSEC course highlights the importance of setting this by describing it as "one of the most important

[4] McClure, Scambray, Kurtz "Hacking Exposed" p87 [23]
[5] Mullen, Timothy M. "An Overview of the Null User" [2]

changes you can make to your system". The key can be found in the following registry location:

HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous

"HKLM" refers to the hive "HKEY_LOCAL_MACHINE". If this key is set to "1" anonymous connections are restricted. An anonymous user can still connect to the IPC$ share, but is restricted as to which information is obtainable through that connection. A value of "1" restricts anonymous users from enumerating SAM accounts and shares. While a value of "2", added in Windows 2000 provides additional restrictions for anonymous connections. This setting has the effect of "No access without explicit anonymous permissions".

However Microsoft [3] recommends that setting the value to "2" should be used in environments with Windows 2000 only, i.e. no mixed-mode environments, as the following tasks will be restricted with this setting:

- Down-level member workstations or servers are not able to set up a netlogon secure channel.
- Down-level domain controllers in trusting domains are not be able to set up a netlogon secure channel.
- Microsoft Windows NT users are not able to change their passwords after they expire. Also, Macintosh users are not able to change their passwords at all.
- The Browser service is not able to retrieve domain lists or server lists from backup browsers, master browsers or domain master browsers that are running on computers with the RestrictAnonymous registry value set to 2. Because of this, any program that relies on the Browser service does not function properly. [3]

As with all recommendations, assurance testing should always be completed in a test environment, to ensure appropriate service levels and required functionality is maintained, before implementing any changes on production systems.

System administrators should not be lulled into a false sense of security, when setting Restrict Anonymous with a value of "1". It is still possible to extract information even with Restrict Anonymous enabled, however a higher skill level and more specialised tools are required. Examples of tools which can be used to bypass the Restrict Anonymous=1 setting are GetAcct [6] and account identification tools user2sid and sid2user by Evgenii Rudnyi [7].

GetAcct has a graphical user interface and one can export results to a comma-separated file for later analysis. It can retrieve user account details and an anonymous connection is not required. One disadvantage of GetAcct

[6] Tool from http://www.securityfriday.com/ToolDownload/GetAcct/getacct_doc.html
[7] Tool from http://www.chem.msu.su/~rudnyi/NT/

is that it does not enumerate the account lockout policy. Therefore, an attacker will have to resort to other means of finding this out if he/she does not want to lock out all of the accounts on a particular domain. One way for an attacker to do this, as described by McClure [25], is password guessing against the disabled Guest account. Although the account is disabled, you will be notified when the account lockout threshold has been reached, with a different error message. From a system administrator's perspective, a countermeasure to this technique is locking out the Guest account, which will in turn stop the account lockout policy from being given away.

User2sid and Sid2user are command line tools that look up the Security Identifier (SID) from username input and vice versa. Once an attacker knows a domain's SID (retrieved using user2sid), accounts can be enumerated using sid2user and appending different RID values, to the already enumerated SID. Consequently, even if an administrator has Restrict Anonymous set, it is still possible for an attacker to find out the name of the user accounts, including the administrator level account. For a more detailed discussion on this process, refer to Packetstorm Security article [28].

2. Weak / Non-existent Lockout Policies

Intruder lockout reduces the chance of unauthorised users "brute forcing" ID/password combinations to gain access to systems. If lockout is not enabled, or lockout settings are not sufficient, attackers have a much higher chance of gaining unauthorised access to systems through guessing user or administrator passwords. It also allows intruders to make use of password "grinding" applications on the network.

According to SANS GSEC [8] recommendations for lockout settings on all domain controllers and sensitive servers / workstations are:

Enable - Account Lockout threshold = 5
Enable - Account Lockout Duration = 30 minutes
Disable - Reset Account Lockout Threshold after

The above recommendations set lockout threshold to five, however it is important for every administrator to implement a setting that best suits his/her organisation. "While you increase the probability of thwarting an unauthorized attack on your organization with account lockout policy, you can also unintentionally lock out authorized users, which can be quite costly for your organization" [9], if you implement an overly strict threshold.

The 'reset account lockout threshold after' setting is often, in my experience, misunderstood and consequently mis-configured. This setting effectively represents the time period in which the number of false passwords allowed or lockout threshold must be met, before the lockout duration will come into

[8] SANS GSEC Section 5.3 "Windows 2000 Security" p29
[9] Microsoft Corporation [21]

effect. I have found that Administrators often make the mistake of setting this lockout duration window to a low number of minutes, which in turn makes the life of the attacker much simpler. For example with a lockout threshold of 5 attempts and a lockout duration window of 10 minutes, you are effectively allowing 4 attempts every 10 minutes. An attacker could easily run a brute - force script, which after 4 attempts, paused for 10 minutes.

Although system administrators may take the time to implement strong lockout policies on domain controllers, often other sensitive servers, such as file servers, are left with no lockout settings at all. It should be noted that the default NT installation has no lockout settings enabled. System administrators rely on the fact that the settings on the domain controller will filter down to the individual servers. However, if one is attempting to logon to a machine with a local account, Windows NT / 2000 will look at the lockout settings on the local machine. This fact is also described in SANS GSEC [10] on the need for setting local security policies, including a lockout policy: "When they are logged on using the local account, security policies set locally will apply. If they are logged on using their domain account, domain policies will apply." For example, when logging onto a machine using a local Administrator account where no lockout has been enabled on this particular machine, unlimited password guessing is possible.

Another hardening measure that can be implemented is enabling PASSPROP.DLL in the Windows Registry. In Windows NT / 2000, the Administrator account (Relative Identifier 500) cannot be locked out, which leads to unlimited password guessing for attackers. However, on Windows NT PASSPROP.DLL allows the administrator account to be locked out remotely, but prevents it from being locked out locally. This dll comes in the NT Resource Kit and once installed, the following syntax at the command line will set Administrator lockout:

passprop /adminlockout

While on Windows 2000, Microsoft provide a utility called admnlock which is a revised version of passprop; it works on machines running SP2 or greater and also requires an account lockout threshold setting to have been set globally for the system / domain. The command line syntax for setting the Administrator lockout is:

admnlock /e

3. Weak / Non-existent Account Policies

"No system is an island in an NT/2000 domain, it only takes one poorly chosen password to unravel the security of your entire Windows environment"[11]. Microsoft [14] also warns that almost any password can

[10] SANS GSEC Section 5.3 "Windows 2000 Security" p18
[11] Scambray, McClure p100 [25]

eventually be cracked by password-cracking software, however the stronger the password the more time it will take. The importance of strong user account policy cannot be underestimated – strong account policies reduce the risk of unauthorised users cracking simple user passwords to gain access to Windows systems.

However, as with the case of Lockout policies, system administrators often rely on the domain controller's settings to filter down to all systems, forgetting that if local accounts exist on a particular computer, then it is the *local* computer settings that take effect. Unfortunately, the default NT/2000 installation, has a minimum password length requirement set to zero, which means it is possible to set a password to be blank or null. Consequently, the *local* Administrator account is often set up with a null password.

However, even more disconcerting is the fact that system administrators, often create *local* accounts on individual servers / systems with the same username and password as their domain accounts. Thus, an attacker who logs on to a non-critical system with a blank password on the Administrator account, can then download the SAM or password file and crack the password that is the same as the domain account password and in so doing can now log on directly to the domain controller, with domain administrator privileges!

I have found that the scenario described above is very common and highlights the necessity for ensuring that strong account policies are enforced across all servers and workstations, as well as domain controllers. An example of strong account policy settings is:

Maximum Password Age = Expires in 90 Days
Minimum Password Age = Allow Changes In 5 Days
Minimum Password Length = At Least 7 Characters* (14 for Administrators)
Password Uniqueness = Remember 13 Passwords

* A discussion on the requirement for 7 characters can be found under "Using LANMAN Password Encryption" point 7 below.

It should also be noted that the practice of creating administrator accounts on *local* machines is not recommended. Rather if the requirement exists, system administrators should specify their domain account (DomainName\Username) as part of the local administrators group on *local* systems and in this way ensure that the domain controller's policy settings are adhered to rather than the *local* machine's policy.

Enabling Passfilt.dll in the Windows registry is also an important step on the road to a strong account policy, as it requires users to select complex passwords. It is included in Microsoft NT Service Pack 2 and should be enabled in accordance with Microsoft's recommendations [4]. However, it should be noted that limitations still exist with Passfilt.dll on NT, as it only filters user requests to change passwords. Administrators can still set weak passwords via console tools, circumventing all of the Passfilt requirements.

Passfilt.dll is no longer required on Windows 2000 as it is installed by default, however it is not enabled. Under 'Local Security Settings' it can be enabled under Security Settings \Account Policies \Password Policy \"Passwords Must Meet Complexity Requirements". Improvements have been made with this Windows 2000 password filter, as this filter applies to all password resets, regardless of whether they are set remotely or from a console.

Joel Kleppinger's article [19] on how to make Windows passwords un-crackable is also an interesting read, where he suggests incorporating ALT-characters into passwords. However although this may be a good idea for administrative level passwords, before forcing this across all users be aware that Helpdesk calls for password resets may well increase!

4. Multiple Trust Relationships

The Windows 2000 Help definition of a trust relationship is a logical relationship established between domains to allow pass-through authentication, in which a trusting domain honours the logon authentications of a trusted domain.

The above definition highlights the inherent security risks associated with trust relationships. Where trust relationships exist between multiple Domains the security of all domains is dependant on the security of the weakest domain in the trust group. Therefore, even if you are confident that your domain has been sufficiently tied-down or hardened, if trust relationships have been established then a weak link into your domain exists. Once a single domain is compromised, administrator or user level access can be obtained on other domains by utilising these trust relationships. In a Microsoft Technet Article [5] on managing trusts, the opening line states that "Trusts require little management", however due to the inbuilt risks that come with trusts, system administrators do need to dedicate some time to manage trusts and ensure that trust relationships are set up with proper security policies and standards implemented on all domains involved. All domains in a trust relationship should have the same security controls and these controls should be those of the most sensitive domain in the relationship. A valid business case should also be in place to justify the existence of all trust relationships between domains.

One of the first steps any system administrator will need to take in managing trusts is to find out what trust relationships actually exist. A powerful tool found in the NT Resource Kit (NTRK)* called 'nltest' enumerates this information, along with more.

* The Windows NT/2000 Resource Kits, provided by Microsoft for administering NT/2000 networks, contain collections of powerful utilities.

However they should not be installed on production systems, "lest the guns be turned against you". [12]

In order to use nltest to enumerate trusts, a null session (anonymous connection) must first be established with a machine in the domain. The following command once run will return details of trusted domains. However, it will only detail the domains that are trusted by the particular domain that the server belongs to, i.e. you will not know whether it is a two way trust or a one way trust.

nltest /server:<server_name> /trusted_domains

As well as enumerating trust relationships, malicious users or hackers will also use nltest to find out the domain controllers in a particular domain. The domain controllers are obviously going to be the primary target for any malicious user, as it is these machines that hold the keys to the kingdom or as described by McClure "domain controllers are the keepers of Windows network authentication credentials". [13] Even if the system administrator is using a system naming convention that does not make it obvious which machines are domain controllers, the following command will enumerate the information for the hacker nonetheless and does not require a null session to be established.

nltest /dclist:<domain_name>

Finally, to enumerate specific user account information hackers also use 'nltest'; once administrator access is gained on a particular domain controller, it is possible to find out details for a specified user, such as when the password was last set and the password hashes. The syntax for this command is:

nltest /server:<server_name> /user:<user_name>

5. Multiple Domain / Local Administrator accounts

One of the key steps in configuring computers for remote administration recommended by CERT is to "Ensure that all administration tasks operate at the minimum necessary privilege level" [14], and according to SANS GSEC "By default, users may have rights that are not needed in order to perform their normal job duties. Enforcing a principle of least privilege for all users is key" [15].

You may wonder why everyone is recommending 'least privilege', after all having just one or two extra administrators with more privileges than they actually need does not have any major security implications and in fact saves

[12] Scambray, McClure, Kurtz p74 [27]
[13] McClure, Scambray, Kurtz p84 [23]
[14] CERT "Configure Computers for Secure Remote Administration" [16]
[15] SANS GSEC Section 5.3 "Windows 2000 Security" p31

time, doesn't it? However, just one of the problem aspects associated with users having unnecessary privileges is that a greater number of domain / local administrator accounts are in existence which in turn means a higher probability exists that one of those accounts can be compromised.

Included in this observation, is the fact that the majority of the time many of these domain / local administrator accounts will not actually be linked to a particular person, i.e. they will be non-owned administrator accounts. These accounts are usually created to support an application or utility, however they still have full ability to access the operating system. In turn these accounts, which are not accountable to a particular employee, are less likely to be properly secured and will be generally set with memorable passwords. In fact these account names generally become public knowledge over time. Many resources are also available for hackers which publish high probability account / password combinations such as the database provided by MK Security Partners [16]. Examples of these high-probability account password combinations are:

Username	Password
Administrator	NULL, password, administrator
Arcserve	arcserve, backup
Test	test, password
Lab	Lab, password
Username	username, company_name
Backup	Backup
Tivoli	Tivoli
Backupexec	Backup

System administrators should assess all administrator level accounts on the domains and individual servers to determine which, if any can be disabled or downgraded to a lower privilege level.

Windows NT / 2000 Resource Kit utilities – local.exe and global.exe, can be used by an attacker to enumerate these accounts, however an anonymous connection is required. The syntax for these commands is:

Local administrators \\IPaddress
and
Global "Domain Admins" \\IPaddress

The tool DumpSec (formerly DumpACL) [17] can also be used to enumerate user and group information. It requires an anonymous connection to the target host, then select "Report" from the menu bar and under "Select Computer" enter the IP address of the target. The reports "Dump Groups as table" and "Dump Users as table" give a lot of valuable information on both groups and users. Attackers can also opt to use the command line version of DumpSec

[16] http://www.mksecure.com/defpw/
[17] Tool from http://www.somarsoft.com

and save the results to a file for analysis. The syntax for this command line option is:

dumpsec /computer= \\IPAddress /rpt=usersonly /saveas=tsv /outfile=c:\temp\users.txt

Systems administrators should carry out a review of all enabled accounts and ownership assigned to an accountable employee (employee who knows the password for the account). Any accounts for which an accountable employee cannot be ascertained, should be deleted or disabled, if not required by a particular service. If more than one employee has knowledge of the account password, the names of each employee should be recorded in the account description field. If one of these employees departs the password to this shared account should be immediately changed.

6. Same Passwords Used Across Domains

In a world where users are required to access multiple applications, on various domains, it is not very surprising that users use the same passwords across these domains. However, administrators who engage in this practice are in turn facilitating the compromise of otherwise secure domains. Passwords that are shared across domains means that privileged user access is often obtainable on other domains by using a cracked password from another compromised system in a different domain. From a hacker's perspective it is the equivalent of having trust relationships between domains. Philip Blow also outlines this risk, in relation to common account names and passwords being used across multiple standalone servers, which cause implicit trusts between the servers [30].

This potential vulnerability could be prevented with a review of the requirement for multiple domains where the users on different domains perform the same role or belong to the same business unit. As well as this, where administrators have accounts on multiple domains they should avoid the use of the same password on each domain, regardless of whether the account is privileged or not. This should be outlined in a security policy document and brought to user's attention through a user awareness program.

7. Using LANMAN Password Encryption

Both Windows NT and Windows 2000 support LANMAN password encryption in order for them to be backward compatible with Windows 98 and Windows 95. However, LANMAN (LM hash) password encryption is weaker than NT encryption (NT hash). Ron Ray in his article describes the results of using LM hashing by saying the "password's strength decreases exponentially" [18]. The reason for this is a LM password is uppercased, padded to 14 characters and divided into two seven-character parts, each of which is used as a key to

[18] Ray, Ron Security Advisory from Securiteam [15]

encrypt a constant. The two hashed results are concatenated and stored as the LM hash, which is stored along with the NT hash in the SAM part of the registry. (On Windows 2000 domain controllers, the password hashes are stored in the Active Directory). Thus, an eight-character password can be interpreted as a seven-character password and a one-character password. Password cracking tools such as L0phtcrack [19] take advantage of this weak design to simultaneously crack both halves as if they were separate passwords. (Randy Franklin Smith's article summarises the process behind L0phtcrack password cracking [20].) Consequently, having a seven-character password is generally more secure than a twelve-character password, as the second half of a twelve-character password would be cracked quicker than the first half, and may contain clues that could aid an attacker in guessing the first half.

Countermeasures to this vulnerability include phasing out the use of Windows 9x machines. Domain Controllers can also be changed to refuse LM password authentication controls. To do this the registry key listed below needs to be located and changed to a setting of 4 on a Windows NT 4.0 system.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\lmcompatibilitylevel

On Windows 2000, this registry setting can be implemented using the Security Policy tool – go to Local Policies\Security Options node with the Group Policy or Local Policy MMC snap in. From here select option to "Send NTLMv2 response only\refuse LM & NTLM authentication"

However, it should be noted that as a result of making this domain controller update on either NT / 2000, a user with an account in that domain will not be able to connect to any member server from a down-level LM client using their domain account. Thus, level 4 means that all users with accounts on a server or domain have to be using Windows NT (SP4) / 2000 to connect and as outlined by McClure "This fix is therefore of limited practical use to most companies that run a diversity of Windows Clients". [20]

The release of Windows 2000 also provided another method of gaining secure authentication with down-level or older Windows clients. By installing Directory Services Client, from the Windows 2000 CD-ROM (dsclient.exe), on older Windows 95/98 machines, system files that provide NTLMv2 authentication are installed. For more detailed, step-by-step information on deploying NTLMv2 on Windows 95/98, refer to Microsoft [6] and [7].

Implementing Syskey (System Key) on Windows NT systems (available from Windows NT 4.0 onwards) also provides an added layer of protection. It establishes a 128-bit cryptographic password encryption key, which is used to create password hashes that are stored in the SAM (Security Account Manager) database. The SAM database is in turn encrypted by the System

[19] Tool from http://www.@stake.com/research/tools/index.html
[20] McClure, Scambray, Kurtz p174 [23]

Key. Administrators can run the command Syskey.exe to set the options for storing the System Key.

There are three Syskey modes or options, as described by Microsoft [8], and they determine the location of the decryption key that protects the SAM database.

1. The key is stored on the local machine in an obfuscated state.
2. The key isn't stored on the local machine at all; instead, the key takes the form of a password that the user must provide when booting the machine.
3. The key is exported from the machine and stored on a floppy disk, which must be presented when booting the machine.

In Windows 2000 and later operating systems, Syskey is enabled automatically, and may not be disabled. Modification options, to move the start-up key to a floppy disk or request password on boot, are available rather than using local storage.

If the Syskey password is forgotten or the Syskey floppy disk is lost, it may not be possible to start the system. Protect and store the Syskey information safely with backup copies in the event of emergency. The only way to recover the system if the Syskey is lost is using a repair disk to restore the registry to a state prior to enabling strong encryption.

However, as described by Philip Cox in his paper on hardening Windows 2000 [9], it is worth considering the threat risk level to a particular machine before deploying a Syskey mode that requires manual intervention. As described above manual intervention is required with two of the Syskey options (inserting a floppy disk or entering password at machine boot) and this in itself can cause management issues and so should really only be deployed for highly sensitive systems. Jennifer Kolde [10] also writes that although storing the password locally is least secure it is actually the most convenient and probably the most suited method for the majority of large organisations.

Finally, although setting Syskey is an important security measure, unfortunately there are ways around this, available to hackers. A utility called pwdump2[21] circumvents Syskey by using DLL injection to load its own code into the highly privileged process space of the Local Security Authority Subsystem (lsass.exe). The code can then access the Syskey-encrypted passwords, however it must be launched interactively on the local machine. Polivec have also released a modified version of pwdump2 called pwdump3[22], which allows for extracting the hashes remotely, by installing the samdump DLL as a service. System administrators can take some solace in the fact that all versions of pwdump require administrator level privileges to run [23].

[21] Tool from http://razor.bindview.com/tools/index.shtml
[22] Tool from http://www.polivec.com/pwdumpdownload.html

8. Auditing Switched Off

A review of the audit logs is the first step in any investigation of unauthorised or unusual activity. Unfortunately, as described in SANS GSEC "Auditing, the recording of security related events, is not turned on by default" [23] and pertains to both the default install of Windows NT and Windows 2000, where all auditing categories are switched off. Hence, the absence of audit logging restricts any analysis of unusual events.

The tool DumpSec (formerly DumpACL) [24] can be used to check whether auditing is switched on or not across a network, or by an attacker. This tool first requires an anonymous connection to the target host, then once connected select "Report" from the menu bar and under "Select Computer" enter the IP address of the target. The report "Dump Policies" will check if auditing is switched on, and will also enumerate the categories of auditing which have been selected.

Both Windows NT and Windows 2000 provide many different categories of security-related events that can be audited. On Windows NT go to User Manager and select Policies | Audit. While on Windows 2000 auditing can be configured through the Local Security Policy (for individual system) or Group Policy (for multiple systems). Naturally, the best would be to enable all of the options but tradeoffs come in huge logs taking up disk space, as well as the amount of analysis time required. It is very important to remember that switching various auditing categories on will not be of very much benefit unless somebody actually analyses the logs. Russ Cooper describes it well when he comments, "just auditing is not enough. Once enabled, you also have to review the event logs regularly and be able to understand what those events mean".[25]

SANS GSEC [26] recommends at a minimum the following should be audited:

1. Account logon events -Both success and failure
2. Logon events
3. Account management
4. Policy Change
5. System Events
6. Object Access - success and failure (files, folders, and registry keys must then be configured for audit. Access to them cannot be configured unless this domain audit policy is enabled.)

The Event Viewer program can be used for event log analysis. It allows for filtering on event date, time, type, source, category, user, computer and event ID, but is still rather time consuming for thorough analysis. Other analysis

[23] SANS GSEC Section 5.3 "Windows 2000 Security" p10
[24] Tool from http://www.somarsoft.com
[25] Cooper, Russ. "Sam Attacks v1.1" [18]
[26] SANS GSEC Section 5.3 "Windows 2000 Security" p30

tools such as dumpel found in the NT Resource Kit, NTLast [27] or DumpEvt [28] are also worth considering for fast and effective event log analysis.

9. Systems not configured with up to date Service Packs

"If the system doesn't have the latest security patches, there is a good chance the attacker will be able to use a known exploit in order to gain additional administrative privileges." [29]

Many Microsoft Windows NT / 2000 Service Packs (SP), security patches and hot fixes are issued to address security vulnerabilities on a regular basis. Failure to maintain systems at current SP levels renders these systems vulnerable to exploitation by hackers using widely available tools.

The "SQL Slammer" or "W32.Slammer" worm, which was released in January of this year, is an example of the disastrous consequences of unpatched systems. "Experts called it the most damaging attack on the Internet in 18 months"[30]. This worm exploited a vulnerability in Microsoft SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000. However, Microsoft released a patch for the vulnerability almost six months ago. The "Code Red" worm released in 2001 is another example of a worm that targeted unpatched systems - in this instance Microsoft IIS servers were the targets.

In an ideal world system administrators would be able to keep all domain controllers, servers, workstations and even test boxes configured with the latest patches, however unfortunately we do not live in an ideal world. As a result, system administrators should aim to prioritise patches, which need to be applied to critical servers and domain controllers. They can do this by keeping up to date with the latest releases and security bulletins from Microsoft's security website [11]. Microsoft also provide a free tool, Network Hot Fix Checker (hfnetchk.exe) [31], which makes the task of keeping up to date with releases a little easier by carrying out checks of which hot fix versions are currently installed and which patches have been applied. Refer to Microsoft [12] for detailed information on hfnetchk.exe.

10. Account named Administrator

The default Administrator account on Windows NT / 2000 systems by default cannot be locked –out by the use of incorrect passwords (However using passprop.dll and admnlock as described earlier does provide this option). Consequently it is the principal target for password guessing or grinding

[27] Tool from http://www.foundstone.com/knowledge/free_tools.html
[28] Tool from http://www.somarsoft.com
[29] Wilson, Zachary "Hacking: The Basics" [29]
[30] Sieberg, Daniel CNN Technology "Computer worm grounds flights, blocks ATMs" [22]
[31] Tool from http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=34935A76-0B20-4F91-A0DE-BAAF969CED2B

attempts and as such, it constitutes an enticement to attackers. Best practice would recommend changing the username on this account. Although this will not stop a malicious user from identifying the Administrator account, it will force them to go a step further in order to identify it (as referenced earlier sid2user/user2sid [32] will enumerate Administrator account from SID) and will also require a higher skill level. For step by step details on how to rename the administrator account reference Microsoft [13].

Scambray and McClure [25] recommend creating a decoy Administrator account and setting it up to look exactly like the true Administrator. In this way if an attacker does not verify that the account is the actual Administrator account by checking the SID, then password-guessing attempts on this decoy account should be easily identifiable in the logs. Remember to enter the standard value in the account Description field, i.e. "Built-in account for administering the computer/domain." As well as this, ensure that this decoy account is not a member of any groups.

Conclusion

In today's world of connected networks the need for security on internal networks is more crucial than ever. The vulnerabilities outlined in this paper represent very common high-risk vulnerabilities, which are found on many internal Windows network today. Although this paper does not encompass all vulnerabilities found on an internal network, it does represent various security weaknesses that an attacker relies upon for privilege escalation, i.e. when an internal network is hacked, the attacker generally does not have to use a glamorous remote exploit to gain Administrator level access, instead he/she will rely upon these weaknesses.

The good news for system administrators is that for practically all of the vulnerabilities discussed there is a solution. By implementing these solutions, the amount of sensitive data given to a malicious user is greatly reduced, as well as avenues for further attack. Greater knowledge of whom and what is on your network is also gained. Using command line tools rather than commercial scanners or scripts that do it all for you, can be time consuming for any system administrator, however it is through these command line tools that true network knowledge is attained, as well as an insight into the tools any hacker will use.

The solutions outlined in this paper need to be implemented on a regular basis. An effective way for a company to do this is to develop an auditing policy, which includes verifying that systems remain secure by checking for changes to systems or unusual logs, on a regular basis. Finally, all administrators need to heed these wise words from SANS GSEC "Security is not a one-time deal but a never-ending process. Just because you are secure today, does not mean that you will be secure tomorrow". [33]

[32] Tool from http://www.chem.msu.su/~rudnyi/NT/
[33] SANS GSEC Section 5.7 "IIS Security" p24

References

[1] Desmond, Paul. "CSI/FBI Security Survey: Questions behind the Numbers" April 30 2002. URL: http://itmanagement.earthweb.com/columns/article.php/1025311 (04 April 2003)

[2] Mullen, Timothy M., "Restrict Anonymous: Enumeration and the Null User". Feb. 12 2001. URL: http://online.securityfocus.com/infocus/1352 (04 April 2003)

[3] Microsoft Corporation. "How to Use the RestrictAnonymous Registry Value in Windows 2000". Microsoft Support Services Website Knowledge Base Article Q246261. 10/10/2002. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;246261 (04 April 2003)

[4] Microsoft Corporation. "How to enable Strong Password Functionality in Windows NT". 6/11/2002 Microsoft Support Services Website Knowledge Base Article ID Q161990. URL: http://support.microsoft.com/?kbid=161990 (24 March 2003)

[5] Microsoft Corporation. "Managing Trusts". Microsoft TechNet Website. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/ad/Windows2000/maintain/opsguide/Part1/ADOgd05.asp (24 March 2003)

[6] Microsoft Corporation. "How to Enable NTLM 2 Authentication for Windows 95/98/2000 and NT". Microsoft Support Services Website Knowledge Base Article Q239869. 10/8/2002. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;239869 (04 April 2003)

[7] Microsoft Corporation. "How to Disable LM Authentication on Windows NT". 10/14/2002. Microsoft Support Services Website Knowledge Base Article Q147706. http://support.microsoft.com/default.aspx?scid=kb;en-us;147706 (04 April 2003)

[8] Microsoft Corporation. "Windows NT System Key Permits Strong Encryption of the SAM", Microsoft Support Services Website Knowledge Base Article Q143475,10/10/2002. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q143475&sd=tech (04 April 2003)

[9] Cox, Philip. "Hardening Windows 2000" 30/3/0.1 URL: www.systemexperts.com/tutors/HardenW2K101.pdf (24 March 2003)

[10] Kolde, Jennifer. "Issues with Syskey in NT 4.0" 3/12/2002. URL: http://www.der-keiler.de/Mailing-Lists/securityfocus/focus-ms/2002-12/0005.html (24 March 2003)

[11] Microsoft Corporation. URL: http://www.microsoft.com/security (04 April 2003)

[12] Microsoft Corporation. "Microsoft Network Security Hotfix Checker (Hfnetchk.exe) Tool is Available" Microsoft Support Services Website Knowledge Base Article Q303215. 18/2/2003. URL: http://support.microsoft.com/?kbid=303215 (06 April 2003)

[13] Microsoft Corporation. "How To: Rename the Administrator and Guest Account in Windows 2000" Support Services Website Knowledge Base Q320053. http://support.microsoft.com/?kbid=320053 (04 April 2003)

[14] Microsoft Corporation. "Strong Passwords". Microsoft TechNet Website. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/server/windows_password_tips.asp (04 April 2003)
[15] Ray, Ron."[NT] Domain Password Logon Authentication Bug in Windows 2000 Advanced Server Domain Controller" Security Advisory from Securiteam 22/7/02. URL: http://www.der-keiler.de/Mailing-Lists/Securiteam/2002-07/0091.html (04 April 2003)
[16] CERT Coordination Center "Configure Computers for Secure Remote Administration". URL: http://all.www.cert.org/security-improvement/practices/p062.html (04 April 2003)
[17] Finamore Joe. "Null Sessions in NT/ 2000". 10/12/2001 URL: http://www.sans.org/rr/win/null.php (04 April 2003)
[18] Cooper, Russ. "Sam Attacks v1.1" 22/7/98 URL: http://www.chebucto.ns.ca/~fifield/jamie/security/docs/SAMAttack.html (04 April 2003)
[19] Kleppinger, Joel. "How to Make Windows 2000 and NT 4 Passwords Uncrackable" 3/01/01. URL: http://sysopt.earthweb.com/articles/win2kpass/ (04 April 2003)
[20] Smith, Randy Franklin. "Cracking User Passwords in Windows 2000". July 6 2000. URL: http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=9186 (04 April 2003)
[21] Microsoft Corporation. "Best Practices". Microsoft TechNet Website. URL:http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsnetserver/proddocs/server/windows_password_protect.asp (04 April 2003)
[22] Sieberg Daniel. "Computer worm grounds flights, blocks ATMs" CNN.com/Technology. 26/1/2003. URL: http://www.cnn.com/2003/TECH/internet/01/25/internet.attack/ (06 April 2003)
[23] McClure, Stuart. Scambray, Joel. Kurtz, George. Hacking Exposed: Network Security Secrets & Solutions Fourth Edition. Osborne/McGraw-Hill 2003.
[24] Microsoft Corporation. "Restricting Information Available to Anonymous Logon Users". Microsoft Support Services Website Knowledge Base Article Q143474. 8/8/2001. URL: http://support.microsoft.com/default.aspx?scid=KB;en-us;q143474 (03 April 2003)
[25] Scambray, Joel. McClure, Stuart. Hacking Exposed Windows 2000: Network Security Secrets & Solutions. Osborne/McGraw-Hill 2001.
[26] Boston, Terry. "The Insider Threat". October 24 2000. URL: http://www.sans.org/rr/securitybasics/insider_threat2.php (04 April 2003)
[27] Scambray, Joel. McClure, Stuart. Kurtz, George. Hacking Exposed: Network Security Secrets & Solutions Second Edition. Osborne/McGraw-Hill 2001.
[28] Packetstorm Security. "Windows NT Security Identifiers" URL: http://packetstormsecurity.nl/NT/docs/sid.htm (06 April 2003)

[29] Wilson, Zachary. "Hacking: The Basics" April 4 2001. URL: http://www.sans.org/rr/toppapers/hack_basics.php (24 March 2003)
[30] Blow, Philip. "IIS Web Servers and Windows Domains ". 30/9/2001. URL: http://www.sans.org/rr/win/domains.php (06 April 2003)