Global Information Assurance Certification Paper

Service Account Vulnerabilities
Barbara Guhanick
8/15/01

Introduction

As an Information Security (IS) specialist, you may be called on by your employer's software application developers to secure an application. This may be a purchased product or a product developed in house. In either case, the earlier an IS specialist is involved in the process, the better since an IS specialist's goals for a product differ from the developers. The application developer's goal is to provide the customer with the product they want and this may mean purchasing or developing a product which has exceptions to your employer's security standards. The developer is most concerned about producing the expected product. An IS specialist is also concerned about producing the expected product, but, one which is within the employer's security standards. Involving an IS specialist early in a project may reduce or eliminate security vulnerabilities by influencing the purchase of a product or how an in-house product's security is designed.

One security vulnerability which arises frequently in application software is the use of "powerful" accounts to either run the application software service, and/or, to be used internally in the application to provide access to data. This paper discusses these special accounts as vulnerabilities in application security in the Microsoft NT/2000 environment.

Service & Application Connector Accounts

Accounts used to run an application software service or process are generally called "service" accounts. An account used to provide access between application software and data is also a type of service account, but, will be called "application connector" account in this paper, since its' role in the application and its' vulnerabilities differ from a service account.

Both of these powerful account types are similar to an interactive user account because they both enable a user to access computer and/or network resources. "A service account is a regular user account with specific properties that allow it to act as part of the operating system."[1]

In a Microsoft Windows Operating System, a Service Account is used to control the access to a service or process. A well known Windows OS service account is SYSTEM. This is an account which is automatically installed with the Microsoft Windows Operating System. SYSTEM is used to run both the operating system, Office and other Microsoft software services and processes. Application software installed in a Microsoft

Windows environment may also require that a service or process be executing. In some instances, the service or process can be started and executed with the authentication and authorization of the SYSTEM account. The SYSTEM account is not vulnerable because it is running as part of the operating system. This is the most secure account with which to run a service or process.

You may find that the software vendor/developer requires that an account with a specific name or with specific access be used to start and execute a service or process. In this case, an account must be created to meet the requirements. Sometimes, this situation arises because software application developers are Windows NT/2000 administrators in their development environment, and if they have not addressed the access needed to run the application in their programming, the resulting application may need NT/2000 administrator access to run.

Sometimes, access to data is streamlined by using a special account to allow communication between application programs and the application database(s). The user may log into the application, but, the access to the data is controlled by the powerful account. The account is used to "connect" the application to the database, and so, for purposes of this paper, will be called an Application Connector Account. The account may also be running a service in addition to being used as an Application Connector account.

Developers have found that for applications requiring user authentication, using one Application Connector account to retrieve data eliminates the complexity of authenticating individual users twice - once when they open the application an again when the data is retrieved. The application programs and database(s) may be on one or multiple servers. The servers may include mainframes.

The Information Security specialist should be aware of the access these "powerful" accounts have, and implement controls to limit the risk created by their use. The IS specialist must recognize the special nature of the service and application connector accounts and look at ways to secure, document and monitor them.

What Makes These Accounts Insecure

These accounts are vulnerable because their IDs and passwords are accessible to both legitimate server and network support and hacker password dumping and cracking tools. With the proper software tools, an ID and password can be discovered. And, once a powerful ID and password are known, an unscrupulous person could affect the performance of the application, server or even the network the server is on. This is type of denial of service attack.

Legitimate and hacker tools have been developed which can read the local SAM and provide information for any service account. "SAM stands for Security Account

Manager. This maintains the security database, stored in the registry under HKLM\SAM. It serves the Local Security Authority (LSA) with SIDs. The SAM maintains the user account database."[2]

In particular, a service account running a service or process is vulnerable because their passwords are stored in clear text. The password can easily be discovered running a free and easy to find on the internet program such as LSAdump2.[3] "This program will dump various LSA secrets such as service passwords (plain text), cached password hashes of the last users to login to a machine, FTP, WEB, etc. plain-text passwords, RAS dial up account names, passwords etc, workstation passwords for domain access, etc."[4]
LSA stands for Local Security Authority. This is an internal subsystem (as opposed to an environmental ditto, such as Win32) within Windows NT that "generates access tokens [...], manages the local security policy, and provides interactive user authentication services."[5]

Once someone has gained local administrator access, free software such as PWDUMP and SAMDUMP can downloaded from both legitimate and hacker internet web sites and be used to collect additional account and password information, open share information and permissions.[6]

Securing A Service Account

There are steps that can be taken to reduce the risk of using a special account as a service account, although the service account remains somewhat vulnerable because of the password being stored in clear text.

- Require a unique ID to run the service on each server it must run on.
- Set up the account as a local server account rather than a global domain account. (Some services need to access other servers, and so, a global domain account would be needed).
- Create a complex password which is at least 8 characters long and uses a combination of letters, numbers and special characters.
- Allow for password aging and expiration if possible. Sometimes service passwords are set to not allow changing and to not expire, because there is a denial of service risk to the production environment if the service account password expires, or if someone changes the password on the account but does not change it on the service.
- Give the account the least amount of access it needs to run the service or process. This includes User Rights, NTFS and Share permissions. Start with the least access and add till the account works.
- Do not share the password. The IS specialist should secure the password in a location which only they or their fellow IS specialists can get to.

- If the password is given to a non-IS person, it should be changed.

Securing an Application Connector Account

The access an Application Connector Account is driven by the requirements of the application. It may need to be a specific type of Microsoft account with specific access requirements. For example:

- Standard SQL or Oracle server account - An application may require a standard SQL or Oracle account with a specific role on the database. This account and password is generally hard-coded in the application software and may require that the account never be deleted or the password never be changed.
- Local Server account - A Windows local account may be all that is needed if the application is contained on one server. The account may or may not be made a local administrator. It could be given specific User Rights and/or authorization to access specific directories or files. The password may be set to not expire and not allow changes.
- Global domain account - A Microsoft Windows Global domain account may be needed. The account may or may not be made a local administrator on the servers involved in the application. It may be given specific User Rights and/or authorization to access specific directories or files across the servers in the application. The password may be set to not expire and not allow changes.
- Mainframe account - An application may require an account to access mainframe databases. In some instances, an ID and password may be created as a global domain account. The ID and password may be hard-coded in the program code to match a mainframe account ID and password in order to provide access across platforms. This usually requires that the account never be deleted or the password changed or expired.
- The account may need a specific name if required by the vendor.

There are various ways to secure the Application Connector account, depending upon what type it is.

- For local and global domain accounts and Mainframe Application Connector accounts, require an ID that is unique to the application across the servers containing the application
- If the account is a standard SQL account, the ID should only be on the SQL server and on the server containing the application
- If the account is connecting an NT server to the mainframe, the ID should be unique to the server and mainframe.
- Give the ID the least access necessary to complete the connection.
- Require complex passwords - include a combination of alphabetic characters, special characters and/or numeric characters. Upper-case and lower-case should

be included, if allowed. Be aware of the special character limitations in SQL and other platforms.
- Ensure any vendor default passwords are changed or removed
- If possible, allow for password expiration and aging. The risk of denial of service because of an expired password should be weighed against the risk of unauthorized use of an ID and password.
- Give the account the least amount of access it needs to function. This access may include User Rights, NTFS and Share permissions, and SQL or Oracle roles. Start with the least access and add till the account works.
- The IS specialist should secure the password in a location which only they or their fellow IS specialists can get to.
- If the password is given to a non-IS person, it should be changed.

Documenting Service and Application Connector accounts

Service accounts and Application Connector accounts tend to be around a long time. A secured list of IDs, passwords, what the accounts are used for, which servers they are on, what access they have been given, who set up the accounts and the set up date is very valuable to an IS specialist. If you are involved in the development or installation of a purchased product, be sure to document these items. Be sure to keep the list update to date by:

- updating the passwords on the list whenever they get changed.
- reviewing the list of accounts on a periodic basis to determine if any can be eliminated.
- removing the accounts from the servers and the list when an application is removed.

Monitoring for Service and Application Connector Accounts

Monitoring the use of the accounts can help in determining whether the account is working correctly and only being used for it's original purpose. An IS specialist can monitor the accounts by reviewing server event logs after account creation to create a base-line showing typical use of the account.
You can also monitor event logs over time to check for account use beyond or outside of the base-line. If you see suspicious account use, take action! This may mean tracking the account use more closely and changing the password if necessary. You will need to determine if additional security needs to be put in place to protect the account.

Conclusion

This paper looked at Service and Application Connector accounts; both special accounts used by application developers to provide their products with access. It covered the security vulnerabilities the special accounts create and the way the special accounts can be secured, documented and monitored. In order to implement these ideas, it is important that an IS specialist be an integral member of the application developer's team. IS specialists can begin this process by sharing this information about special accounts with application developers they work with. Both IS specialists and application developers must work together to produce the best result; a secure product.

List of References

[1] Microsoft Corp., Using a Service Account
http://msdn.microsoft.com/library/en-us/biztalks/htm/lat_secure_intro_pwtl.asp?frame=true

[2] Simple Nomad ,The Unofficial NT Hack FAQ, Section 05
http://www.fastlane.net/homepages/thegnome/faqs/nt/

[3] Paul Ashton, NT LSA Secrets
http://www.insecure.org/sploits/NT.LSA.secrets.html

[4]Robert Malmgren, Copyright @ 1996, NT Security Frequently Asked Questions Version 0.41, 2.1.9
http://www.rito.com/nt/ntsec/faq/sec-faq.htm#pw-service

[5]Windows NT resource guide, ISBN 1-55615-653-7

[6]Evadenet, Password Hash Extraction
http://www.evadenet.com/downloads/lophtcrack.shtml