Global Information Assurance Certification Paper

SANS/GIAC Practical Assignment
For GSEC Certification
By Garo Doudian
Version 1.4b
Dated: June 7, 2004
Option 1, Research on Topics in Information Security
16 pages total

Generic Accounts and Non-Repudiation

The capability, in security systems, that guarantees that a message or data can be proven to have originated from a specific person. [1]

Table of Contents

Purpose

The purpose of this paper is to discuss the importance of non-repudiation in a corporate environment, focusing on generic user accounts. Non-repudiation is defined as "*The capability, in security systems, that guarantees that a message or data can be proven to have originated from a specific person.*"[1] Various methods will be analyzed, along with each of their benefits and drawbacks.

Non-repudiation is a necessity in everything from e-commerce to systems management. As new laws start requiring it, non-repudiation will become more important to companies. This paper will be looking at non-repudiation from a systems administration and account perspective. Putting measures in place to ensure that a set of policies and infrastructure can properly audit events is key to achieving non-repudiation.

Introduction

As computers play a larger part in everyday corporate life, physical and digital identities are becoming increasingly intermixed. In both worlds, knowing who performed certain actions is very important. Improper identification can lead to a variety of problems including open audit issues, wrongful termination, and even lawsuits. With stakes like these, it's no wonder that companies want to do everything in their power to reduce ambiguity whenever possible.

The proper tracking of generic system or service accounts, and who is using them, is a crucial step in reducing the unknown factor in the corporate world. Although generic accounts are, by their nature anonymous, there are ways that companies can protect themselves from possible litigation and monetary loss. Through hardware, software, and corporate policies, the impact of generic accounts can be greatly controlled.

Individual Administrative Accounts

Most operating systems, applications servers, and databases use generic system and service accounts. Generic accounts are system IDs that aren't assigned to an individual user. They are often built-in accounts used to configure and manage the systems as well as manage other accounts. In many situations, multiple people may know the ID and password for these accounts, and no one account owner exists. In the event of an emergency, or even daily processes, these top level accounts should be available to several authorized people in order to eliminate any single points of failure. Unfortunately, the use of generic accounts by more than one person is a direct violation of non-repudiation. So, how does one use generic accounts to administer systems in a way that allows

for accurate auditing and tracking, while still maintaining a high level of functionality?

The simplest method of reducing the use of generic accounts is to grant individual administrators higher levels of access. With enhanced access, IT support staff can perform privileged tasks without resorting to the use of the generic accounts. Since each account is tied to one unique individual, proper audit logs can be generated. Support personnel would have these administrative accounts as well as their own regular user account. Following the Principle of Least Privilege, which dictates that "*a user be given no more privilege than necessary to perform a job*"[2], it is a good idea to only grant higher access when needed. Therefore, only the administrative account for each support person would have access to privileged functions. The regular IDs would be used for all daily tasks, such as Internet usage, email, and other common user functions. Using regular accounts for daily tasks, instead of the higher level ones, will cut down the number of mistakes that may occur during regular usage. The administrative accounts, on the other hand, should only be used when a task requires elevated privileges. These accounts shouldn't be left logged on, and no common user functions should be available to them.

It's a good idea to use a special naming convention when creating these accounts. For example, users jsmith and tfranklin would have administrative accounts named admin_jsmith and admin_tfranklin. Having a standard naming convention also makes monitoring easier, as well as reducing confusion during log evaluation. Since only administrative tasks would be used with these accounts, auditing levels would be greatly increased and all activity recorded in detail. These higher level functions may be used to grant permissions, change system-wide settings, or even add and remove users. Likewise, these accounts may be used to restrict access to sensitive data that should not be available to others. Lastly, logging all changes to these administrative IDs, and their respective administrative groups ensures that no unauthorized changes to the administrative accounts themselves are made. Instituting these measures is insufficient if a user is able to create a generic administrative account to perform unauthorized actions.

Having generic account and password information available to many people will increase the chances that someone will intentionally or unintentionally leak out sensitive information. Compromised, or leaked, account details can be very devastating for a company especially since generic accounts tend to have privileges that most standard users do not. It is important to realize how crucial the security of these accounts and password details are. A trade secret that gets leaked to the Internet can be quite damaging, especially if the source of the leak isn't identifiable. Listed below are just a few of the possible ramifications of having high level generic account details compromised:

"*Why should you care?*

- *Damages prestige of the University [or company]*
- *Bad press directly/indirectly influences:*
    - *Faculty, students, staff and alumni*
    - *Potential faculty, students and staff*
- *Causes us to become a known target*
- *Weak security = easy target*
- *Word gets around VERY QUICKLY in hackerdom*
- *Worst case*
- *New York Times front page article deriding you, your department and the University"[3].*

Reducing the number of people who have access to these high level accounts, along with proper auditing will help prevent possible password sharing incidences, especially since people are more likely to share information about an account that isn't directly tied to them.

Forms of Authentication

In the above cases, the assumption is made that the generic accounts in question can be replaced with non-generic equivalents. However, that is not always possible. Consider the case of the local administrator accounts on Windows systems. These built-in accounts cannot be removed, and are crucial to the operation of the systems. Although it may be possible to create individual local accounts for each support employee, this is highly impractical. Since each local account is specific to one machine, a separate account would have to be created for each admin, for each server. An environment with 50 servers and 8 support staff members would require a total of 400 separate accounts. There are times, however, where using a generic account for certain administrative tasks is a necessity. Consider what happens if, in the above example, a Windows machine somehow falls off of the NT or Active Directory domain. In order for it to be rejoined properly a local administrator would have to log onto the machine, and re-add the machine to the domain. Without having a local account for each support person, the built-in Administrator password would have to be known by anyone who may take an emergency call like this.

Many applications have a generic service account that the software uses to operate. Microsoft SQL and Exchange servers are two examples of such applications. Tying these service accounts to an individual's ID is deemed a bad practice for several reasons. First of all, the application may need more access rights than the user does, and following the Principle of Least Privilege, giving the user this enhanced access is not allowed. Furthermore, the password change policies may be drastically different for users and service accounts. Frequent password changes are more common for end user accounts than for service accounts. Using separate logins for these applications makes more sense from an operations standpoint, but because they are not tied back to an individual, they should be treated like generic accounts.

So if generic accounts have to be used, how can they be reliably tracked and how can we still maintain a high level of security and traceability? The solution to this dilemma can be found by first evaluating the three main types of authentication factors.

- Something you know
  - Passwords, passphrases
  - Secret questions/answers
  - Personal information
- Something you have
  - Physical token
  - ID Card
  - USB key
- Something you are
  - Biometrics, fingerprints, retinal scans, etc.

Tokens – The physical authentication factor

Most companies use "Something you know" in the form of a password. Unfortunately, something you know may be something that others know as well. While somewhat secure, this form of authentication also doesn't prevent the sharing of account details, thereby granting access to other individuals. WindowsSecurity.com has published an interesting article discussing how passwords are oftentimes the weak point in an organization. "*The password, then, functions like the key to a lock; anyone who has it can get it. This means the password can easily become the weak link in a company's network security plan, because passwords can be "cracked," guessed, stolen or deliberately shared. It is important for individual users to safeguard their passwords and for organizations to educate users regarding best password practices and to develop policies that mandate that such practices be followed."[4]* Since it is easy for a password to leak out in an organization, a good practice is to add another level of security so that passwords aren't the only method of gaining access to systems and information.

Tokens, smart cards, software certificates, and other physical devices fit into the "something you have" category. These devices cannot be easily copied; therefore only one person will have the proper authentication piece to log onto an account. Many companies, like RSA and Securikey, sell these physical devices. Using "something you have" along with "something you know" gives you "*the same level of security that makes your ATM card secure. The two factors are: "something you know" and "something you have". In the case of your ATM card, the "something you have" is the plastic card and the "something you know" is your PIN. And now for your computer system, the "something you know" is your password and the "something you have" is your SecuriKey USB token*"[5]

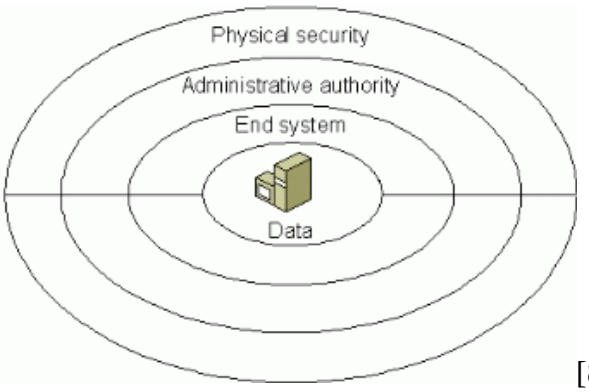
So using tokens, along with passwords, to validate generic account logons is much safer than using passwords alone since there are now two separate layers of required authentication. The most popular and widely used tokens in the market today come from RSA. Their SecurID product is designed to complement the existing password-based infrastructure while adding in a series of new benefits. The use of these tokens greatly increases the level of non-repudiation that we can achieve as well as providing many more security features, some of which are listed below:

"*Relative Security*
• Two-factor authentication results in very strong form of security.
• Passcodes are generated dynamically and are less vulnerable to cracking tools.
• Passcode changes every 60 seconds, eliminating the threat of visual theft of passcodes and Trojan horse threats.
• Passcodes cannot be guessed or predicted.
• Users are aware when a token is stolen or lost.
• Network transmission of token codes cannot be easily detected.
• Improves security by eliminating the need to write down passwords.
• RSA ACE/Server® software provides logging and reporting functionality for greater end-user accountability.
• Centralized administration eliminates security holes as new devices, applications and communication methods are added and users are added, deleted or change roles. "[6]

As you can see, tokens allow administrators to combine various independent security models into one robust, integrated approach without limiting the operational ease of use of the systems. Using tokens also adopts a Defense-in-Depth strategy. The Enterprise Security Corporation states that in a Defense-in-Depth design, "l*ayers are very important because they develop a layered security posture and add levels of protection and containment of vulnerabilities*"[7]

"*When building and designing a Defense In-Depth strategy keep the following basic points in mind:*
** Each layer is only aware of itself*
** Each layer of your Defense In-Depth strategy must be auditable*
** Each layer supports local and centralized logging*
** Each layer subscribes to a synchronized time source (NTP)*
** Put the layers together to build the big picture*"[7]

"*A security-in-depth approach first divides all security elements into discrete security layers. This way, the security effectiveness of each layer can be independently determined, and a security plan can be implemented*"[8]



[8]

In the above diagram each individual layer adds to the overall protection of the data. Using tokens, in addition to passwords, enhances the security model by adding in physical security controls to password measures already in place. As more and more independent methods of authorization are added, the data contained within becomes better protected. Technically, even more authentication factors, aside from tokens and passwords, could be used to build on this model resulting in a more secure solution.

Usage tracking with tokens

One of the most common applications for tokens is to validate users over VPN connections. Since VPN users connect remotely, the lack of physical security, by allowing logins outside of the corporate secured buildings, can be compensated for by using tokens. Since the token is a material object, physical security measures can still play an important role. Tracking who has the token at any point in time allows for accountability.

Aside from VPNs, token technology can also be used to authenticate for other systems. It is possible, for example, to link a windows login account with a token. If a generic account that is tied to a token logs in at a certain time, the only person who would have been able to connect would be the person with the token. In this way, an account that was previously completely generic can now be accurately audited, and its actions can be traced back to one individual. The RSA SecurID tokens use a complex algorithm to generate a seemingly random series of numbers. The only two entities that know a given number sequence at any moment in time are the login server, and the token itself. Therefore it is highly unlikely that anyone other than the person with the token would have the proper credentials to log into the protected account.

There are however several downsides to using tokens. First of all, a separate token is needed for each administrator and/or generic account used. For larger companies with hundreds of computers and many generic accounts this added cost can be burdensome. Secondly, most generic accounts should rarely, if ever, be logged in. Again, take the local administrator accounts on Windows machines as an example. These accounts should only be used when

first installing the Operating System, associated applications, and joining the domain. Once the machine is part of a domain, user specific privileged accounts should be used for all administrative purposes. With this in mind, why invest in all these tokens when they would rarely be needed? Aside from the cost of tokens, the fact that there could possibly be dozens, or even hundreds, of physical objects that now need to be secured, logged, and monitored could cause its own problem. This approach simply trades digital security problems for physical ones. Where would all of these tokens be kept? It now becomes clear that using tokens for many rarely used generic accounts isn't always a very practical solution.

Password Vaults - Physical

One solution to the token problem comes in the form of password vaults. A password vault is a secure location where account passwords, usually generic, are stored until they are needed. Much like a vault at a bank, authentication steps are put in place to ensure that only authorized people have access to the data contained within it. A password vault can be either a digital or physical entity. In the physical approach, generic user account details are printed out on paper and sealed, usually by lamination. The sealed pieces of paper are then stored in a physical safe, locked drawer, or another secured area. Usually the initial generic or system account password is generated automatically, or in pieces by more than one person. It is essential that no one knows these passwords unless they have looked it up on the piece of paper. When a password is needed, it is retrieved from the vault and opened.

Although this solution has solved the problem of securing the passwords, it's important to audit this access, otherwise there is no non-repudiation. There are several ways to audit access to the password vault, but it's important to consider all the benefits, and shortcomings, with each alternative before adopting a solution. For example, passwords could be sealed in various folders and secured inside of a locked room. Only one person would have access to this room and they would require that all access be logged. When an authorized administrator needs a password, this person would log all the details, and hand over the requested information. At first glance this seems like a highly secure solution, but in reality it isn't. Not only does this solution rely on a manual process, which in itself is inherently unreliable, but it also gives one person complete access to the vault. This access may not even be audited properly since it is this individual's sole responsibility to keep the audit logs.

Another solution might be to have this same room controlled by an automated badge access. Any user needing access to the room would use their badge to open the door. All audit logs are automated, and since they aren't relying on a person to record them, they will be more reliable. Unfortunately, this solution isn't much better than the previous one. Although this approach addresses both issues raised with the last solution, careful examination shows

that it's not secure after all. This system would audit when a user accessed the vault, but it wouldn't be able to log what they accessed once they were inside. Of course once the seal on an envelope has been broken, it becomes obvious that someone accessed the password but the actual specifics of who and when would be difficult to log. Obviously this too, doesn't achieve a reliable method for non-repudiation.

Taking into account the shortcomings of the previous examples, a more secure solution can be designed. All access to the vault itself is audited electronically by badge readers. This outer security point would validate that the person has any business in the vault itself. It also serves as a central point of authentication. This outer perimeter could also have a guard or support person watching. One layer of auditing and alerting could be done here as well. Once inside the vault, each password would be sealed in its own "safety deposit box" where it would only be accessible by another badge reader. Again, this access would be audited and monitored. Once someone has accessed the password they need, the paper would be re-sealed, and locked back up. The corresponding audit records would also be recorded detailing who accessed the password, and at what time. For enhanced security, each of these steps can use a two factor authentication method, and several people can grant or deny a request. Because everything is audited, at any given time a record can be printed detailing who knows each password, along with when they were accessed.

The drawback of physical solutions is that they tend to be more costly to implement, as well as maintain. They also usually require some sort of manual processing, be it checking video tape, or guarding a door, which has a tendency to be unreliable. For information that should be highly accurate, this unreliability factor may be too high. However, physical solutions do have some benefits as well. They can easily take advantage of other security measures already in place throughout the enterprise. For example, security cameras that may be in place already can also be used to watch the vault. Likewise, security personnel would have the means to check the vault on their normal watch route. Furthermore, in a disaster recovery situation, duplicating a physical solution may take less time than a digital one.

Password Vaults - Digital

Although physical solutions can be used to secure access to data, it makes more sense to use a digital solution to store digital data. Like physical solutions, digital solutions can provide an effective and more practical way to secure passwords, while maintaining a high degree of auditing. The most effective way to secure passwords is by using a digital password vault. A password vault is a secured application running on a system that stores the details of any desired account. Like its physical counterpart, this "digital vault" should have appropriate physical security considerations. A good example

would be a vault running on a server in a Data Center, locked behind a cage, under video surveillance and monitored by authorized staff members. The vault application server should also be heavily protected from the network. Again following the Principle of Least Privilege, all server services and functionality not directly involved with securing data are disabled or removed. The vault server may also have a firewall installed that prevents any access to it except for the few authorized methods required to access the vault.

The vault administrator(s) create individual storage areas for each item that they want to protect. These storage areas would be akin to a safety deposit box within a bank vault. Access is then given to the required support personnel. Whenever a password is needed, the support staff, using the vault application, would connect to the vault, authenticate, and retrieve the desired information. Again authentication plays a critical role in this solution, so many different factors can be used. First of all, like many other systems, a user is given a unique account and password, or pass-phrase. Since each user would have their own individual account on the vault server, there is no problem with generic account tracking. Generic accounts should never be used to connect to the password vault for normal operation because it would defeat the purpose of the vault, which exists to eliminate multiple people from knowing generic account details. These individual accounts can be tied into pre-established authentication measures, like a Windows domain. Tying in to an existing authentication method is beneficial in several ways. First, there is not much extra effort in maintaining these accounts, since they would already have the proper support infrastructure to monitor and track usage. Secondly, associating the password with another system reduces the number of individual passwords a support staff member needs to memorize. The less they need to remember, the stricter the policy and enforcement can be on each administrative account. Since most of the information stored in the password vault may be for rarely used generic accounts, administrators may not access the vault for months at a time. Allowing them to authenticate with the vault using a password that they use daily for other tasks will reduce the number of times that a user forgets his/her password. If administrators had a unique password that they never used anywhere else, chances are high that they would forget this password when it came time to use the vault, making the data contained within inaccessible.

Looking back at the various factors of authentication, additional layers can be added by requiring the use of secure tokens, or smart cards. Depending on the criticality of the data stored in the safe, the information can also be restricted so that two or more authorized persons must clear an access request before the account details can be retrieved. Multi-factor authentication provides more security at the cost of some ease of use, but is important for highly sensitive data. User accounts can also be restricted to logon from certain machines or certain times of the day, thereby tying in all existing physical security measures. The level of security should be directly proportional to the sensitivity and criticality

of the data stored in the vault. The most secure pieces of data would probably contain all of the measures listed above, while the least could use just a few.

Once someone has authenticated themselves to the vault, they will be shown only what they have access to. This idea of only knowing about data elements that you're authorized to see is similar to the Bell-Lapadula model. The Bell-Lapadula model is "*the most famous model of protection systems [that] deals with the control of information flow. It is a linear non-discretionary model. This model of protection consists of the following components:*

- *A set of subjects, a set of objects, and an access control matrix.*
- *Several ordered security levels. Each subject and object is assigned to its own security level.*
  In this case, we use (unclassified < confidential < secret < top-secret)

The security levels are used to determine appropriate access rights. The essence of the model can be defined as follows:

- *A higher-level subject (eg. secret level) can always "read-down" to the objects with level which is either equal (eg. secret level) or lower (eg. confidential / unclassified level). So the system high (top security level in the system) have the read-only access right to all the objects in the entire system.*

- *A lower-level subject can never "read-up" to the higher-level objects, as the model suggests that these objects do not have enough clearance to read the high security level information.* "[9]

By not even allowing unauthorized users to see what data is stored in other safes enhance overall security. Each user's view of the vault should be completely customized to show them only what they need to see. The ability to customize what users see is another large benefit of using a digital solution over a physical one.

Considerations

The more forms of authentication that are used, the more security and non-repudiation is achieved, but at the cost of less operability. The types of authentication methods that can be required are almost limitless and could be scaled up or down depending on the criticality of the data contained within a safe. Once the user has properly authenticated to the vault, they would have access to the information that they need. Whenever information is retrieved or modified, it must be audited and monitored. It is critical that everything is audited, and that nobody, not even the head system administrators, can delete the audit logs. Many commercially available products have a minimum log age that can be set,

and once configured, no entry can be deleted by anyone unless it is older than the minimum history retention age.

Digital solutions are more prevalent in the corporate world today because they provide better functionality in an automated manner. Whenever possible, human dependencies should be removed, and handled instead by automated system functions. In order for any solution to work in the corporate world it must be reliable enough to provide the auditing information that administrators demand, a high level of security, and a level of operational ease of use that makes it a practical implementation.

These few examples were a good start in addressing the problem of generic account password storage. They touched on some key points that should always be taken into consideration when designing a solution. For any implementation to be practical, the following must be addressed:

1. **Secured location** – First and foremost it's important to make sure that unauthorized users don't have any access to the safe. Logically, a safe should be a great deal harder to break into than the systems that they are protecting. Otherwise, every hacker would go straight for the safe since it holds the skeleton keys to the other systems. A secured location can be physically secured, digitally secured, or both.

2. **Auditing** – Once someone has validated their access, it is crucial to record this event so that it can be reviewed at a future date. In most organizations this is the most important piece of the whole puzzle, since audit logs are the only way to tell what happened in the event of a break in. Proper logging also provides reliable evidence, which could be critical if any criminal or civil charges are ever pursued, and as such should always be closely guarded and impossible to erase. A whole new paper could be written on this topic alone, so let's just say that securing audit logs is a mission critical focus.

3. **Monitoring/Alerting** – Unlike auditing, monitoring is used for more real-time data collection. Alerts are an important way to prevent any security incidences before they get out of hand. Certain events should be configured to send an email, page, call, or even fax a notification group as soon as they occur. If someone accesses the vault in the middle of the night, it may be important that someone be notified immediately instead of waiting for the event to be discovered in the audit logs the next day or week.

4. **Functionality** – It's always important to balance the level of security with the level of functionality in any solution. For example, the most functional solution to a doorway would be to have it completely open with no door so that people could come and go as they please. On the

other side of the coin, the most secure doorway would be to seal it off with concrete so that nobody could ever get in. Obviously neither of these solutions would be what anyone would implement for their front doors, but rather, a hybrid of the two is best. When choosing a password vault mechanism it's important to keep in mind how much security and functionality are required. If the passwords are for disaster recovery, and should only be used in the event of a catastrophic failure, then it's not unreasonable to balance the equation so that it may be less functional and more secure. On the other hand, if the vault is accessed on a daily basis, it would be beneficial to design the solution so that it doesn't take 2 hours to get to the needed information.

Policies and Procedures

Once everyone agrees on a solution, sets aside the money and the time to build it, and actually implement it, the company has a great deal of interest in keeping the solution operational. With this in mind, it's critical to put together policies and procedures to prevent anyone from circumventing the new process. For example, it's quite common for system administrators to share passwords with one another whenever a need arises. A new staff member may ask a coworker what the generic login information is for a certain system. Usually, both people are authorized to know the passwords since they are there to support the associated systems, but this can pose a real problem when it comes to auditing. A simple question to ask yourself is, if someone were to come to you and ask you right now, "Who knows the password for account ABC?" would you know the answer? How sure would you be?

When it comes down to it, most companies are seriously lacking in this sort of auditing. Even though each support person is a trusted individual, and they each trust each other, it's important to prevent anyone from unilaterally sharing password information as this action becomes nearly impossible to log. With a password vault solution in place, the proper people are given access to each piece of information that they need to perform their jobs, but they may not always know each generic account password. Whenever they need a password they would go to the vault and retrieve it themselves. This way an accurate list can be created at any time showing who knows which password(s). Likewise it's important to audit this access list, and compare it to the "real world" if possible. Generic account usage should be monitored to ensure that only people who are listed as knowing the password know the password. If only one person is listed as knowing a certain password, and another person is found using it, you have a problem. The best solution in the world won't prevent passwords from spreading unless a good, strong policy is drafted, and strictly enforced. Understandably, this process can be seen as an extra hurdle in the way of the support personnel since they now have to log into the vault to get the necessary information, but

without it, everything that was done to create this amazingly secure safe is undone in a matter of seconds.

Another important factor to keep in mind is that once more than one person has accessed a particular generic account password, determining which of the two people may use the account becomes a problem again. As a result, frequent password changes are also critical to keeping this solution secure. Certain disaster recovery accounts, for example, should only be used in emergencies. If, in the middle of the night, a support person needed the password, they would go to the vault, get the password, and do what they needed to do. Once the administrative task has been completed, it's important to take the next step the following day and change the password for this account. This ensures that the next time that account is needed, and another person accesses the vault and retrieves the password, only one person will know the account information and there are no issues with non-repudiation. While this could cause more work upfront, it will greatly reduce exposure, and reduce the potential risk of security incidents. Each generic account that is used in emergencies should have a unique, difficult password that's hard to memorize. In an environment with hundreds of servers this could mean having hundreds of different local administrator account passwords but this is a requirement if you wish to secure your environment properly.

Passwords should be generated and stored in the vault automatically, or by someone who sets a random password without memorizing it. An easy way to set passwords manually would be to have two people come up with half of the password and type it in without letting the other person know what they typed in. After entering each half of the password, the two people would destroy the paper that contains their half. This way no one person ever knows any of the passwords. As with the other policies, this password generating policy is a critical piece of the solution.

Summary

Generic accounts bring with them many inherent problems and risks. Although sometimes it's possible to use other means of administration, there are many instances where this is not a viable alternative. There will always be times where companies are forced to use generic accounts for various administrative and operational tasks. By considering the issues discussed in this paper, companies can develop a plan of action that will minimize their exposure to these risks. Providing a method of accountability for generic account usage either by individual administrative accounts, tokens, password vaults, or a combination of these, the company can develop a system that will allow for greater security and ease of use. Combined with a good set of polices and robust auditing, this infrastructure yields a solution that will be a viable one for years to come. The problem of generic accounts shouldn't keep the systems administrators up at night. Though each solution on its own may have drawbacks, adapting the

various methods to match what the company is looking for will present an answer to the generic account problem. If things are designed right, every single generic account's actions will be linked back, without a doubt, to one user, finally achieving non-repudiation.

References

[1] Alliance for Telecommunications Industry Solutions - Wed Feb 28 15:39:21 MST 2001
http://www.atis.org/tg2k/_nonrepudiation.html

[2] John Barkley – Mon Jan 9, 1995
http://hissa.ncsl.nist.gov/rbac/paper/node5.html

[3]Scaring you secure - Marc DeBonis - IS&C – Virginia Tech – 2001 – Page 8
http://security.vt.edu/gotoclass/scaringsecure.pdf

[4] Windowssecurity.com – Security Topics – Passwords: The weak link in Network Security – Deb Shinder - May 07, 2003
http://www.windowsecurity.com/articles/Passwords_Network_Security.html

[5] Think Geek – Security Authentication systems – Securiky USB tokens
http://www.thinkgeek.com/gadgets/security/5cd6/

[6] RSA – The Authentication Scorecard, page 10 – © 2003
http://www.rsasecurity.com/products/authentication/whitepapers/ASC_WP_0403.pdf

[7] Enterprise Security Corporation – Defense In Depth – © 2000-2002
http://www.esecuritycorp.com/topstrategies/defenseindepth.htm

[8]Computer Security Models – Eddy Chan – May 29, 2001
http://infoeng.ee.ic.ac.uk/~malikz/surprise2001/spc99e/article1/

[9]Microsoft - Best Practice Guide for Securing Active Directory Installations and Day-to-Day Operations: Part I - By Kathleen Cole, Dave Kreitler, Doug Steen, and Markus Vilcinskas – Updated February 28, 2004
http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/maintain/bpguide/part1/adsecp1.mspx

Last Updated: July 28th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event