Case Study: Least Privileged Administration in a Windows Environment – NT vs. Active Directory

William Hicks

GIAC Security Essentials Certification
Version 1.4b, Option 2
August 9, 2004

Abstract

Windows NT made it difficult to implement a widely accepted security principle called *Least Privileged Administration* to manage the degree of system rights granted to support staff. As a result, privileges on NT4 domains are essentially granted on an “all or nothing” basis. In one company, which is the subject of this case study, that limitation contributed to having large numbers of administrators with excessive system privileges.

The case study examines some ramifications of a flawed NT4 security model and lax security administration practices, and then describes interim steps implemented to work toward Least Privileged Administration in an NT4 environment. Security delegation features (and some limitations) offered via Windows 2000 Active Directory are presented next. I conclude with a discussion of how a vendor solution improves upon native Windows 2000 delegation tools, and how it was used to set up security delegation in an Active Directory to achieve a tighter security model that aligns with the principle of Least Privileged Administration.

Background: Poorly Managed NT Domains with Too Many Administrators

I was hired in early 2000 as a security administrator at a company with 24,000 employees, which was growing at the rate of 1,000 per month. Unfortunately, as the company grew explosively in terms of employees, systems administrators, and infrastructure, sound security controls sometimes take a back seat to other company priorities, like production, customer-focus, and availability. Of course as security professionals, we recognize the inherent short-sightedness of this approach to managing a network. Familiarizing myself with this new environment, several issues concerned me about its security posture.

- **Large numbers of Domain Admins**. Shortly after hiring on, I audited the headquarters NT domain and identified 225 Domain Admins, to support approximately 10,000 users and 13,000 hosts. I found 13 active accounts with Domain Admins rights that belonged to former employees. Company-wide, I discovered there were just over 600 Domain Admins to manage 20 domains. Due to a complex NT domain topology and existence of common infrastructure (such as e-mail and proprietary business systems), a relatively large number of administrators had Domain Admins rights in multiple domains. The goal of information security is to protect confidentiality, integrity, and availability of technology systems (SANS, 15). Having large numbers of Domain Admins increases the likelihood of breaching all three. Domain Admins have complete access to data (potentially impacting confidentiality and integrity) and to systems (a risk to maintaining availability).

- **No formal authorization to obtain privileged systems access**. Newly hired NT administrators were normally granted Domain Admins credentials regardless of job role, training, or experience. Admin IDs were created by peers, with no procedures to confirm validity or obtain authorization for the account.
- **Poor audit practices for privileged accounts**. Consistent with the philosophy of Least Privileged Administration (LPA), it is common practice to maintain separate "admin accounts" for work requiring higher privilege, and use unprivileged accounts for all other functions. In many cases, our administrators' unprivileged account had the same group memberships as their admin account, not only undermining the rationale for having separate accounts, but also contributing to the high number of accounts with Domain Admins rights. This evolved over time because no one audited or managed privileged group memberships.

It is my belief the situation was symptomatic of two deeper problems, which I'll discuss in greater detail below.

- Security policies were not well socialized, and there were few published security procedures
- Administrative delegation capabilities in NT4 are limited

Security Policies Not Socialized, A Dearth of Security Procedures.
In their discussion of high level security policy objectives, the IT Security Cookbook states there are several "Critical Success Factors" for implementing good policy, including: "Comprehensive guidance on security policy and standards must be distributed to all employees and contractors." (sect. 6.0)

Our technology organization, in contrast to others I'd previously worked for, does not require new systems administrators to receive formal training about corporate policies and procedures, including those related to security. As a result, there was often "plausible deniability" when administrators were challenged about actions which violated the letter (or the spirit) of security policy.

Additionally, a number of security *policies* existed but were vague with respect to implementation details. Unfortunately, documented security *procedures* were virtually non-existent, and imprecise policy language allowed wide latitude for administrators to implement security as they saw fit. Security *policy* differs from security *procedure* in that policies are high level documents which provide guidance about "who, what, and why" certain security measures are enacted, whereas procedures address implementation details, including "where, how and when" (SANS, 86).

Administrative delegation in NT4 is limited. Although several built-in groups created by NT4 during domain controller setup permit some delegation to manage user and machine accounts, servers, and printers, it is non-granular (i.e.

can not be further sub-divided), and exists only on domain controllers. Member servers and workstations only provide for delegation to “Power Users”, and “Backup Operators”, whose inherent rights are pitifully inadequate to perform most administrative tasks. A more in-depth discussion of NT4’s built-in groups is available at: <http://www.microsoft.com/technet/prodtechnol/winntas/maintain/acctgrps.mspx>.

Microsoft defines *Least Privileged Administration* (LPA) as: “A recommended security practice in which every user is provided with only the minimum privileges needed to accomplish the tasks they are authorized to perform, and no others.” (Microsoft, “Glossary”). Ironically, they designed NT4 in a way which makes implementation of LPA nearly impossible. Installing an application? You need to be an Administrator. Connecting to that new color printer for the first time? Oops, need an Administrator to load those drivers. And had security patching been considered as important by many systems administrators back in the NT4 days as it is today, they’d have been out of luck without (you guessed it), Administrator rights.

Interim Steps in NT4

Five other NT administrators were hired at the same time I was (in early 2000) to address Windows security concerns in each of our regional service centers. I asked my colleagues to perform the same analyses in their own domains that I had done at headquarters to establish a baseline. As we audited and documented findings, administration practices were found to be “consistently inconsistent” company-wide. We catalogued security concerns and began brainstorming possible solutions, to prioritize those issues which we had power to influence, and those which would have greatest impact on overall Windows security.

Overcoming NT operating system limitations would be difficult, but the team could educate management about risks related to poor procedural controls, and begin solving those first. We focused on two areas: authorizing and managing administrative privileges, and auditing privileged access.

Authorizing and Managing Administrative Privileges.
For his SANS practical assignment, Mark Austin wrote eloquently about the problem of having too many Domain Admins. It was his experience that people who had these rights would go to great lengths, what he called “Machiavellian maneuverings”, to maintain them. Austin and other senior administrators in his organization overcome the problem by interviewing their staff and basing decisions about admin privileges on job function, rather than job title or MCSE status.

Revoking admin rights from staff that should not have them would be difficult, “taking away” from people something they valued. Thus, we had to balance ensuring administrators had sufficient rights to do their jobs, but avoid creating hostility or disrupting work in the process.

My team used an approach similar to Austin's – we surveyed our Windows administrators to determine what tasks each functional group/team performed. By analyzing those tasks, we could objectively determine what sets of rights and permissions were necessary to complete them. Besides the obvious fact-finding rationale for surveying our staff, I believed revoking improper or excessive system privileges would be better received if it were conducted via open and participatory process. Our survey revealed two general problem areas: *improper* and *excessive* administrative rights. These findings were consistent with the two underlying factors I described earlier: *inadequate security procedures*, and *poor NT delegation capabilities*.

Improper administrative rights: By fixing two procedural problems, we could immediately reduce the number of Domain Admins by 35%.

- Revoke the credentials from staff whose job function did not involve administration of the domain. Generally, folks in this category supported only small numbers of member servers or workstations, and never supported domain controllers or managed user accounts. Their improper privileges were not the result of NT's poor delegation capabilities, but rather from inadequate understanding of NT security and lack of proper authorization procedures.
- Revoke Domain Admins from what *should* have been unprivileged accounts. This sloppy practice had evolved out of convenience, since the "Run-As" feature of Windows 2000 did not yet exist. Technicians did not like closing applications and authenticating with their admin account to perform privileged tasks. Without sound authorization processes to manage which accounts had Domain Admins privileges, the widespread practice continued unchallenged. It exposed the company to unnecessary risk because e-mail and other applications were associated with user accounts that had global admin rights. Imagine the havoc malware, such as mass mailers or network worms can cause if executed with an Administrator's security context.

Our team recommended to management that we revoke these improper rights. We would also document authorization process and criteria necessary to obtain privileged access to Windows domains. We based those criteria on performance of tasks identified in the survey as those which truly required Domain Admins privileges. After receiving management approval, and with documentation to support our actions, we revoked Domain Admins rights from just over 200 accounts.

Excessive administrative rights: Organizationally, our Windows administrators were classified as Server Support or Desktop Support. We analyzed distinctions between tasks performed by Desktop Support and Server Support teams. A subset of the desktop support function included our corporate Help Desk. As expected, many people who provided desktop support had Domain Admins credentials. However, the tasks they routinely performed did not support those rights, even with NT4. We learned through the survey and via follow-up interviews that resolving user account lockouts, home and profile permission issues, imaging and joining workstations to the domain, and solving desktop application problems comprised the majority of their ticket load. On the other hand, people who provided server support for the most part legitimately required Domain Admins rights.

Focusing on the Desktop Support function, we could downgrade rights through a few relatively simple changes. For example, unlocking user accounts and joining workstations to a domain require Account Operators membership, but not Domain Admins (Microsoft, "Understanding User Accounts"). We also knew that the built-in Domain Admins global group facilitated management of the domain's servers and workstations because it is automatically added to the local Administrators group when computers join a domain. We created a plan to put this knowledge to use. By following the steps outlined below, we incremented closer to LPA while still constrained by NT4 domains.

- Create new global groups, which contained Help Desk and Desktop Support staff. Using separate global groups for each team facilitated greater access control granularity.
- Add these new global groups to each appropriate domain's Account Operators local group, which enables password reset and workstation account privileges. Of course, Account Operators also permits unwanted privileges, especially ability to create user accounts, but we were limited by NT4's capabilities to delegate further.
- Script a deployment of the new global groups to the Administrators local group of all NT4 hosts managed by the Help Desk and Desktop Support teams.
- Update desktop images to ensure these new global groups were added to Administrators when new computers were built and joined to domains.
- Publish and socialize a security guideline, prohibiting unauthorized account creation by those who had ability, but not authority to do so.
- Modify file permissions on home/profile directories, allowing Help Desk and Desktop Support teams sufficient permissions to manage the directories without requiring Administrators rights to those servers.
- Help Desk and Desktop Support staff could then safely be removed from Domain Admins.

We also built an exception process whereby members of these teams who had to perform one-off tasks that required elevated privileges could attain them for a specific period of time. We piloted a transition with senior staff from each team. Once satisfied they could complete their jobs seamlessly, the Desktop Support staff was less reluctant to relinquish higher privileges. Because we'd proved the concept through a pilot, management was comfortable we could revoke Domain Admins from Help Desk and Desktop Support staff without impacting production availability. They readily threw their support behind the initiative, and we phased it over a three month window to further reduce the likelihood of creating a production issue.

The NT security administration team, which I was then leading, was one year into its existence. We had reduced the number of Domain Admins from 620 to just over 150 company-wide, without creating a single production outage, and with only minimal restructuring to the way technicians performed their tasks. For the first time, the company had measurable controls over privileged access to its Windows environments.

Auditing Privileged Accounts. I knew efforts we'd taken to reduce rogue user accounts and admin rights would erode over time without additional controls. Procedural documentation without effective monitoring is useless, especially with well-entrenched practices we were trying to reverse. I asked one of my teammates to create a Perl script which pulled specific events related to new account creation and global group membership changes from domain controller's security logs. I also engaged the server monitoring unit to activate a feature of our NetIQ system which would alert my team's on-call pager when sensitive group membership changes, (i.e. groups which conferred privileged access), were made. These two controls ensured new user account creation, and changes to sensitive groups were detectable and auditable. Unauthorized changes were made rather frequently at first; after all "that's how it was always done before." However, we saw a steep decline shortly after implementing monitoring, because detection of these events was nearly real-time. Our team consistently and quickly addressed incidents with the responsible individuals. Word out: someone is now watching and holding administrators accountable to published security procedures.

Active Directory / Windows 2000 Security Delegation

The successes of my team had earned company-wide visibility, and I was invited to join a small team to develop the security delegation model for our Active Directory ("AD"). The delegation team's charter was to create a scalable and flexible security model based on Least Privileged Administration (LPA) to accommodate ongoing organizational changes as the company evolved. To achieve our objective, we first had to study the mechanics of Windows 2000 delegation. For the sake of completeness, I'll note here the scope of our project also extended to researching Group Policy Object (GPO) design and implementation, but that is outside the scope of this paper.

What is delegation? According to Microsoft's Delegation Best Practices manual, "Delegation of administration is the transfer of administrative responsibility for a specific administrative task from a higher authority to a lower authority." It classifies system administration tasks as *service* oriented or *data* oriented. Service administration involves AD's core infrastructure and operability (e.g., security, DNS, replication, schema, and domain controller management), whereas data administration is concerned with managing resources and data contained in the AD (e.g., user and computer accounts, Organization Units, and data stored on AD hosts). Distinguishing between service and data administration makes it easier to identify and separately delegate rights required by the few highly skilled engineers who design and maintain an AD infrastructure from other administrators, whose focus is commonly more local, i.e. at a departmental level, in large organizations like ours. Our primary research resources were Microsoft architectural and implementation White Papers for Windows 2000, and from FastLane, one of our vendors. The Microsoft White Papers are no longer available on their website, however through the power of Google, they can still be downloaded from: <http://www.barsonconsulting.com/StudentCenter/W2kwp.htm>. Microsoft has of course updated their web site's reference materials to focus on Active Directory engineering and implementation using Windows 2003. Since Windows 2000 and 2003 are functionally similar as far as delegation is concerned, I'll continue to reference "Windows 2000" in this paper, as it was the OS our project worked with. The company has since upgraded its AD forest to Windows 2003.

How does delegation work? To discuss delegation, I first need to define a few terms. In essence, Windows 2000 treats Users, Computers, Groups, and a new construct (beginning with Windows 2000) called ***Organizational Units*** as ***objects*** in a directory, which is itself essentially a large extensible database. In its AD Architecture documentation, Microsoft defines Organizational Units (OUs) as logical boundaries within domains, typically implemented to subdivide them for the purposes of administrative control, i.e. to create delegation boundaries. It defines an Object as, "…a distinct, named set of attributes that represents something concrete, such as a user, a printer, or an application." The ***schema*** defines properties of all ***object classes*** (groups of objects) and their associated ***attributes***. All object classes have numerous distinct attributes defined by the schema, and each object has the same attributes as all other objects in its class (Microsoft, "AD Schema"). The schema also specifies allowable ***rights*** on each attribute, such as Create, Read, Modify, and Delete (Mar-Elia). Using a perhaps more familiar terminology, one could say AD ***objects*** represent database ***records***, and ***attributes*** are similar to the records' ***fields***.

Mar-Elia, a widely published author in Windows circles, states delegation is possible in Windows 2000 partly because access to all AD objects and attributes is managed through a discretionary access control list (DACL). Object DACLs in Windows 2000 operate very much like those on NTFS file systems. By manipulating the ACL Editor (accessed via Properties\Security tab of AD objects), one may control which operations can be performed by individuals (or more preferably, by members of ***security groups***). Security groups are groups which are used to "…grant rights and permissions to users" (Microsoft, "AD

Architecture”). As with NTFS, permissions to manipulate a given attribute or object can be granted at different levels to various security groups, i.e. Administrators have Full Control, while others may only have Read or Modify permissions on the same attribute. This feature enables Windows 2000 to offer flexible, though complex security configuration options.

Another property of Windows 2000’s security design that enables delegation to work is inheritance (Mar-Elia). As with DACLs, the inheritance properties used to manage delegation are similar in function to their NTFS counterpart. Permissions on objects which are lower in the AD hierarchy are inherited by default from their parent, so they do not have to be individually set at each layer. Inheritance simplifies delegation; in most cases security is configured at OUs, and is allowed to propagate downward to like objects within that delegation boundary. Inheritance can be blocked when it becomes necessary to create a unique permission structure on child objects, which provides additional security flexibility. However, inheritance blocking should be applied judiciously, as it will complicate future delegation tasks.

Delegation in action: NT4 vs. Windows 2000. Let’s assume Help Desk staff is authorized to change user account passwords and unlock accounts, but does not have any other authority relative to user accounts. In an NT4 domain, only Domain Admins and Account Operators can create, modify, or delete user accounts. There is no way to delegate this authority to any other user or group (Mar-Elia). Furthermore, all Domain Admins and Account Operators can perform the same actions on all accounts as everyone else with the same credential. How does this relate to LPA? An administrator whose *authority* is limited to unlocking user accounts or resetting its password still has *ability* to modify other properties of the account, e.g. change its group memberships. S/he may also create or delete user and computer accounts by virtue of the same credential. In other words, with NT4 ability to affect a user account is absolute, or it is non-existent. Therefore, LPA is not maintained, since the administrator has more system ability than authority.

In Windows 2000, one can now permit Help Desk staff ability to *only* unlock or reset passwords on user accounts; permissions to all other user attributes can be set to Read, and fields on the user account can even be hidden from members of specified security groups. Help Desk’s password reset ability can be delegated to all user accounts in the domain, or to any subset. This is all achieved without membership in Domain Admins or Account Operators, significantly reducing security exposure from excessive system privileges. Thus, through proper delegation, LPA can be realized.

Windows 2000 delegation challenges. There is great power in being able to so specifically control how administrators can manipulate each AD object, but this flexibility also presents some challenges. According to Mar-Elia, there are “…something like 190 different types of objects and over 1000 attributes out of the box…” He continues, “…the possible permutations for securing the AD are staggering.” Microsoft’s native Delegation of Control Wizard offers a simple interface through which delegation can be applied to any object or group of objects, but it has some limitations.

- Microsoft’s Delegation of Control wizard and ACL Editor are “stateless”, which MarElia describes as “…no central notion of what the security on an object should be.” In other words, an administrator with sufficient privilege can modify permissions set by a previous administrator, and because of default inheritance properties, s/he can effect a large and perhaps unintended change on child objects. Without good reporting, an unintended or unwise security change may go undetected on child objects, depending on inheritance configuration.
- The Delegation of Control Wizard and ACL Editor cannot store security templates, i.e. settings which can be reused at other delegation boundaries (Mar-Elia). For example, assume you have several teams of support staff who manage their own OUs, and you must delegate ability to add computers to their OU, but to no others. According to Microsoft’s delegation best practices manual, the appropriate approach in most cases is to use the Delegation Wizard at each target OU. However, because the wizard must be invoked at each OU, in large environments with multiple delegated tasks, this is tedious work.
- Using the native ACL Editor, one can determine *effective permissions* of a specific individual or security group on a named object, but it’s not feasible without scripted or third-party utilities to do this on large numbers of objects, or for the AD as a whole.[1]

Delegation Pilot and Implementation Using ActiveRoles

Vendor solution to delegation challenges. Since our company had already purchased a third-party product to facilitate NT4 migration to Windows 2000, our delegation team had an opportunity to evaluate FastLane’s ActiveRoles as we developed our model. This product is now called *ActiveRoles Direct*, as Quest has since acquired FastLane. Mar-Elia states that ActiveRoles addresses the limitations of Microsoft’s native delegation tools in the following ways:

- *Ability to create Roles.* Roles are essentially “templates” to facilitate security delegation. Roles save the DACLs needed to control operations on objects. If all delegation tasks were easily defined by single well-known attributes, this would not be such a great benefit. However many delegation tasks involve complex DACLs applied to groups of attributes. Since attributes are usually not intuitively named, and some amount of

[1] The Effective Permission feature was not available in Windows 2000, so this limitation was an even more serious concern at time. Effective Permissions functionality was added to Windows XP and Windows 2003.

research is needed to determine which attributes must be modified to ensure proper delegation, ability to save and reuse these settings becomes more clearly beneficial.

- *Enforce proper security settings*. Although any administrator with proper credentials can still alter permissions of a Roles-managed object, this action will result in an easily detectable mismatch via the product's user interface. Once detected, the anomaly can be corrected by re-applying the Role. More proactively, ActiveRoles can be scheduled to periodically search for permission mismatches, which can either be automatically corrected or simply alerted through its reporting mechanism.
- *Flexibility*. Once Roles are defined and applied, when changes become necessary, one only has to modify the single Role, and the new permissions automatically propagate to all objects managed by that Role. Using Microsoft's native delegation tools, one would have to reapply that change at each delegation boundary.

AD pilot. Now armed with knowledge about the mechanics of Windows 2000 delegation, delegation best practices from Microsoft and FastLane, and a tool we believed would be useful, our delegation team set up shop. We built a separate pilot AD forest, which mimicked the structure of the forest being developed for production.

The basic methodology of our project is outlined below.

1) Research systems administrator's job functions – what tasks each team (e.g., Help Desk, Desktop Support, Server Support) performed in order to maintain their systems.
2) Determine which tasks were suitably delegable. For example, ability to fully manage workstation OS is not delegated from the AD; the task simply requires membership in its local Administrators group.
3) Of the delegable tasks, determine which object classes (e.g., users, groups, or computers), attributes (e.g., object name, user password, group membership), and operations (e.g., create, modify, delete or take ownership) are associated with those tasks.
4) Determine appropriate delegation boundaries for each task (e.g., OU, child OU or domain).
5) Create security groups containing each organizational/functional team's technicians.
6) Create Roles that set appropriate DACLs on managed objects. Bind Roles and security groups containing appropriate administrators to proper delegation boundaries.
7) Systems administrators test ability to perform tasks in pilot forest.
8) Document the delegation scheme. Submit documentation and Roles to the AD Design / Engineering team for peer review and approval.
9) Migrate Roles from pilot to production AD forest; test and monitor.

To begin, our team modified and reused the Domain Admins survey discussed previously to assess tasks performed by various teams of administrators. Since I already had some experience from the NT4 Domain Admins reduction effort, the survey was assigned to me. I have attached a sanitized version of this survey as Appendix A. Using survey results, I built a matrix of tasks, teams, and controls needed to manage or delegate those tasks.

With documentation from Microsoft and FastLane (now Quest), we spent considerable time researching which of the thousands of object attributes related to common systems administration tasks. At the time, this type of documentation was not nearly as available as it is today, so we partnered with FastLane and Microsoft engineers. Our team tested several dozen pre-built Roles which were packaged with ActiveRoles. These were essentially modules of commonly delegated tasks. The pre-built roles helped us develop a more practical understanding of which attributes and operations to delegate in order to achieve a particular outcome. We were able to use several without modification, though it is easy to customize the pre-built Roles, and create new ones from scratch.

Some of the pre-built Roles we evaluated for User delegation included: *Create User*, *Modify User*, and *Reset User Password*. When multiple delegated tasks are performed by members of the same security group, several pre-built Roles can be combined and saved as one. This feature greatly simplifies Role implementation, since they are applied at different delegation boundaries. Besides being a time saver, compared to using only native delegation tools, reuse of the Roles also ensures consistency.

In the end, our team employed a “keep it simple” approach for the delegation scheme. We generally set the same permissions on all attributes of an object class for a given security group. In other words, the team responsible for maintaining the user account lifecycle was given Full Control of the entire user object in most OUs of our AD. When warranted due to organizational structure, we sub-delegated to other teams ability to update certain user attributes, such as password reset to the Help Desk, home/profile path to desktop support staff, etc. This allowed those teams to continue performing tasks historically assigned to them, without having ability to modify any other User attributes.

Ability to create and delete security groups was granted exclusively to our account management team in order to maintain consistency in group naming and hierarchy (i.e. nesting). However, ability to manage most groups’ members was delegated to appropriate teams, again allowing them to maintain an historical degree of autonomy over that responsibility.

Significantly improving our ability to control privileged access to the AD, we designed for a separate locked-down OU to contain administrator accounts and security groups which are bound to Roles that delegated privileges. Only members of my Windows security administration team had ability to create or modify user accounts and security groups within this OU. The delegation team also created Roles to lock down privileged built-in groups, such as Domain Admins, Administrators, and Enterprise Admins, to name a few. This achieved something impossible to do with NT4 – even a Domain Admin is unable to add other accounts to that built-in group unless they override the Role to which is bound. This unauthorized action would be very quickly detectable (and correctable) with ActiveRoles.

Once we created Roles to control delegation for each set of system administration tasks identified in the survey, we bound them to appropriate OUs in our pilot forest, and performed extensive testing to determine precisely what actions were allowed (and denied). This task fell primarily to me, since I was the delegation team's most experienced Windows technician. Once satisfied the Roles were functioning as intended, we engaged a team of volunteer systems administrators from the field to perform additional QA, intended to test their ability to perform appropriate (as well as inappropriate) administration tasks in our pilot environment.

After testing, we submitted documentation of the delegation strategy, Roles definitions and settings, and test results to the AD design team and management for peer review and final approval. Roles were then migrated from our pilot forest to the production AD. Since Roles can be exported in file format, the migration was accomplished with minimal effort, and demonstrates again a benefit of using ActiveRoles compared to using the native delegation wizard.

Since the production AD had not yet received significant numbers of migrated users and computers from NT4 domains, we were confident any differences between the pilot and production environments would not have significant operational impact if Roles were still imperfect. The real test came as migration efforts began later in 2001. Our security delegation team was an integral part of the AD migration process in order to have timely and first hand knowledge of issues that arose from our delegation strategy. As one can imagine, we encountered several situations which were not anticipated or tested during the pilot. When these problems arose, the delegation team regrouped to find a solution; in most cases a minor tweak to the production Roles ensured technicians had proper ability to support their environments and clients.

Summary

In a two year span, I served on two teams which made significant progress toward achieving Least Privileged Administration in our Windows environment. Where NT4 domains had 600+ Domain Admins, our AD currently has fewer than a dozen permanent members to support the corporate forest. Could that number be made smaller? In a perfect world yes – further refinement of the Roles could probably result in another 50% reduction, but ever-present organizational and political considerations make that unfeasible at this time. Using the FastLane tool, similar reductions in the number of Account Operators and Server Operators were also achieved with the Windows 2000 migration. While it may be difficult to qualitatively measure improvement to the company's security posture which resulted from these initiatives, from a quantitative standpoint, the numbers speak for themselves: with 50 times fewer administrators having unbridled access to the entire Windows infrastructure, the potential attack surface available for insider malfeasance is unquestionably smaller. From a procedures perspective, controls were for the first time implemented to authorize and manage administrative privileges, and automated audits alert security administrators when rare exceptions occur. These procedures are well socialized among systems administrators, and more often than not, they seek permission from security before taking action, rather than forgiveness afterward. Finally, we designed the security of our AD in such a way that future organizational adjustments can be easily dealt with– one can apply or remove Roles as easily as making security group membership changes.

Perhaps if people were always inclined to do the right thing, and were not subject to making mistakes even as they tried to do so, Least Privileged Administration would not be such an important cornerstone of information security. Since that will never be so, it is incumbent upon us as security professionals to ensure systems users and administrators have procedures and guidelines help them make sound decisions, but to also back those up with good controls to guard against the foibles of human nature.

Works Cited

Austin, Mark. *The Divine Right of Kings: Domain Administrators and your (In)secure Network.* August 17, 2001. SANS. August 3, 2004. URL: <http://www.sans.org/rr/papers/index.php?id=306>

"IT Security Cookbook". Updated July 28, 2000. Boran Consulting. August 5, 2004. URL: <http://www.boran.com/security/>

Mar-Elia, Darren. *Simplifying AD Security with Quest Software's Microsoft Solutions*. Updated February, 2004. Quest Software. August 3, 2004, URL: <http://www.quest.com/content/list.asp?ContentTypeID=1&GroupBy=CategoryName&Format=table&nav=/activeroles/navigation.xml&prod=42&productfamily=ms+management>

"Active Directory Schema." Updated May 2004. Microsoft Corporation. August 17, 2004. URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adschema/adschema/active_directory_schema.asp>

"Microsoft Security Advisor Program: Glossary of Terms." Microsoft Corporation. August 11, 2004. URL: <http://www.microsoft.com/technet/security/bulletin/glossary.mspx>

SANS Institute. *Track 1 – SANS Security Essentials and the CISSP 10 Domains. Defense In-Depth*. SANS, January 2004.

"Understanding User Accounts, Groups, Domains, and Trust Relationships." Updated April 1, 1998. Microsoft Corporation. August 11, 2004. URL: <http://www.microsoft.com/technet/prodtechnol/winntas/maintain/acctgrps.mspx>

"Windows 2000 Server Active Directory Architecture." 2000. Microsoft Corporation. August 11, 2004. URL: <http://www.barsonconsulting.com/StudentCenter/W2kwp.htm>

"Windows Server 2003 Best Practices for Delegating Active Directory Administration." Updated November 24, 2003. Microsoft Corporation. August 6, 2004. URL: <http://www.microsoft.com/downloads/details.aspx?familyid=631747a3-79e1-48fa-9730-dae7c0a1d6d3&displaylang=en>

Appendix A – Delegation Survey

NT Administrative Security Assessment

This assessment form will be used by designated security officers and management to evaluate the NT administrative security level you will be given based on the support requirements you have for specific server and workstation resources. Our goal is to provide technicians with the right amount of access so that they can get their job done, yet minimize the security risks for the company. Please answer the questions below as accurately as possible and return the form to the sender. The security officers will review the information, contact you if adjustments need to be made, answer any questions you may have about the changes, and then work with you to make the process as smooth as possible. Please return the form by the date designated by the sender. If you have any questions, please contact .

Your Information

Name: **Extension:**

Department: **Location:**

<u>Primary</u> Job Role: (check one only)

□ NT Server System Administrator (NT Level 2 Support)
□ NT Server and Desktop (Workstation) System Administrator (NT Level 2 Support)
□ NT Desktop (Workstation) System Administrator (NT Level 2 Support)
□ Service Desk System Administrator / Technician (NT Level 1.5 Support)
□ Service Desk System Administrator / Technician (NT Level 1.0 Support)
□ NT D&E
□ NT Level 3 Support Print
□ NT Level 3 Support Exchange
□ NT Level 3 Support NT
□ Software Distribution (Marimba) Admins
□ Developer & System Administrator for a Development Group
□ UNIX System Administrator
□ Netware System Administrator
□ ISS
□ Voice System Administrator
□ Network System Administrator (DNS Admins)
□ Access Management Administrator
□ Other. Explain below:

(Enter for a new line)

I. Server Support Requirements

Appendix A – Delegation Survey

A) What NT Servers are you required to support? "Support", in this context, means that you need to add, configure, or remove any logical or physical component of the server to ensure that the resources are available to the end-users. Components include:

- Hardware components and configuration of these components
- Operating System services, directories, and files
- Supporting administrative software, services, directories, and files
- End-User Application services, directories, or files (including NT Policies)
- Security configuration
- User home shares, directories, and files
- User profile shares and files
- Public shares, directories, and files
- Printer queues
- Web / FTP setup/management

If you do support NT Servers, list the server names, business groups that use the resources on the servers, and the location of the servers. If you support an entire domain of servers, simply indicate the domain and business groups that use the domain resources, and the location of the PDC for the domain. Please include all domains/servers that you support, regardless of the domain role (resource domain as opposed to a master account domain).

□ I am **not** required to support NT Member Servers.
□ I am required to support NT Member Servers. The servers/domains, related business groups, and locations are supplied below:

Server or Domain Names	**Related Business Groups**	**Location of Servers or PDC**

(Enter for a new line)

□ I am **not** required to support NT Domain Controllers.
□ I am required to support NT Domain Controllers. The servers/domains, related business groups, and locations are supplied below:

Server or Domain Names	**Related Business Groups**	**Location of Servers or PDC**

(Enter for a new line)

B) Are you required to add, remove, or modify hardware components on the servers you support?

□ No. I am not required to add, remove, or modify hardware components on servers.
□ Yes. I am required to add, remove, and modify hardware components on servers.

Comments (optional)

(Enter for a new line)

C) Are you required to install or reinstall the NT Operating System on the NT Servers you support?

□ No. I am not required to install or reinstall the NT Operating System on servers.
□ Yes. I am required to install and reinstall the NT Operating System on Servers as needed.

Servers and comments

(Enter for a new line)

D) Are you required to **upgrade** NT Server Operating System components on the servers you support (such as device drivers, service packs, or firmware/BIOS updates)? Which servers?

□ No. I am not required to update these components.
□ Yes. I am required to update these components as required.

Servers and comments

(Enter for a new line)

E) Are you required to **modify** Operating System Settings (network configuration, device configuration, pagefile settings, performance settings, start-up/shut-down settings) on the servers you support?

□ No. I do not need to modify Operating System Settings.
□ Yes. I do need to modify Operating System Settings when needed.

Comments (optional)

(Enter for a new line)

F) Are you required to add or remove NT servers in any of the domains you support? Which domains?

□ No. I am not required to add or remove servers in any domain.
□ Yes. I am required to add and remove NT servers in the domains that I support. The domains are:

Domains and comments

(Enter for a new line)

G) Are you required to change share-level or object-level security (i.e., directories and files) on **server-based** resources? Explain.

□ No. I am not required to change share-level or object-level permissions on servers.
□ Yes. I am required only to change share-level and object-level permissions on user's home directories, or public directory structures on the servers I support.
□ Yes. I am required to change share-level and object-level permissions on Application directory structures, along with user and public directories on the servers I support.

□ Yes. I am required to change share-level and object-level permissions on **any** share or object on the NT Servers I support.

Comments

(Enter for a new line)

H) Are you required to add, move, modify, or delete files or directories on the servers that you support?

□ No. I am not required to add, move, modify, or delete files on servers.
□ Yes. I am required to add, move, and delete files in User home directories, Profile directories, or Public directories only on the servers I support.
□ Yes. I am required to add, move, modify, and delete files in the Application Directory Structures only on the servers I support.
□ Yes. I am required to add, move, modify, and delete files in any share or directory on the servers I support.

Comments

(Enter for a new line)

I) Are you required to add, remove, or modify **software** or services on the servers you support?

□ No. I am not required to add, remove, or modify software on servers.
□ Yes. I am required to add, remove, and modify software on the servers I support.

Comments (optional)

(Enter for a new line)

J) Are you required to stop and start services on the servers you support?

□ No. I am not required to stop or start services on servers.
□ Yes. I am only required to stop and start services on the servers I support.
□ Yes. I am required to install, remove, configure, stop, and start services on the servers I support.

Comments (optional)

(Enter for a new line)

K) Are you required to have the ability to reboot (locally or remotely) any of the NT Servers you support?

□ No. I am not required to have the ability to reboot servers.
□ Yes. I am required to have the ability to reboot servers when needed.

Comments (optional)

(Enter for a new line)

L) Are you required to backup or restore files to the NT Servers you support?

□ No. I am not required to backup or restore file to NT Servers.
□ Yes. I am required to have the ability to backup and restore files only to user's home directories or public directories on the servers I support.
□ Yes. I am required to have the ability to backup and restore files on any part of the file system on the servers I support.
□ Yes. My primary job is to backup and restore files to NT Servers as requested.

Comments (optional)

(Enter for a new line)

M) Are you required to add, configure, manage, or remove Printers (print queues) on the NT Servers you support?

□ No. I am not required to add, configure, manage, or remove printers on servers.
□ Yes. I am required only to manage Printers on the NT Servers I support (this includes deleting or restarting documents, or stopping and starting the Spooler service).
□ Yes. I am required to add, configure, manage, and remove printers as needed.

Comments (optional)

(Enter for a new line)

N) Are you required to log-on **interactively** (locally) to the NT Servers?

□ No. I am not required to logon interactively to NT Servers.
□ Yes. I am required to logon interactively to the NT Servers I support.

Comments (optional)

(Enter for a new line)

O) Are you required to view Event Logs on the NT Servers you support?

□ No. I am not required to view the Event Logs on servers.
□ Yes. I am required to view the Event Logs on the servers I support.

Comments (optional)

(Enter for a new line)

II. Workstation Support Requirements

A) What NT Workstations are you required to support? "Support", in this context, means that you need to add, configure, or remove any logical or physical component on the workstation to ensure that the resources are available to the end-users. Components include:

- Hardware
- Operating System services, directories, and files
- Supporting administrative software, services, directories, and files
- End-User Application services, directories, or files
- Security configuration
- User profile files

If you do support NT Workstations, list the business groups that use the workstations. If you support an entire domain of workstations, simply indicate the domain and business groups that use the domain resources.

□ I am **not** required to support NT Workstations.
□ I am required to support NT Servers. The servers/domains, related business groups, and locations are supplied below:

Business Groups and/or Domain

(Enter for a new line)

B) Are you required to add, remove, or modify hardware devices on the workstations you support?

□ No. I am not required to add, remove, or modify hardware in workstations.
□ Yes. I am required to add, remove, and modify hardware devices in the workstations I support.

Comments (optional)

(Enter for a new line)

C) Are you required to **image/re-image** (or install/re-install the NT Operating System on) the workstations you support?

□ No. I am not required to image or install the NT Operating System on workstations.
□ Yes. I am required to image / re-image the workstations I support when needed.

Comments (optional)

(Enter for a new line)

D) Are you required to add or remove NT workstations in any of the domains you support? Which domains?

□ No. I am not required to add or remove workstations from domains.
□ Yes. I am only required to ADD workstations to domains.
□ Yes. I am required to ADD and REMOVE workstations to/from domains.

Comments (optional)

(Enter for a new line)

E) Are you required to **upgrade** NT Workstation Operating System components on the workstations you support (such as device drivers, service packs, or firmware/BIOS updates)?

□ No. I am not required to upgrade NT Workstation Operating System components.
□ Yes. I am required to upgrade NT Workstation Operating System components when needed.

Comments (optional)

(Enter for a new line)

F) Are you required to **modify** Operating System Settings (network configuration, device configuration, pagefile settings, performance settings, start-up/shut-down settings) on the workstations you support?

□ No. I do not need to modify Operating System Settings.
□ Yes. I do need to modify Operating System Settings when needed.

Comments (optional)

(Enter for a new line)

G) Are you required to add, modify, or remove files or registry entries on the workstations you support?

□ No. I am not required to add, modify, or remove files or registry entries on workstations
□ Yes. I am required to add, modify, and remove files and registry entries on the workstations I support.

Comments (optional)

(Enter for a new line)

H) Are you required to remotely stop and start services on the workstations you support?

□ No. I am not required to stop or start services on workstations.
□ Yes. I am required to start and stop services on remote workstations.
□ Yes. I am required to install, remove, configure, start, and stop services on remote workstations.

Comments (optional)

(Enter for a new line)

I) Are you required to have the ability to reboot (locally or remotely) any of the NT Workstations you support?

□ No. I am not required to reboot [end-users] workstations.
□ Yes. I am required to reboot (locally or remotely) workstations that I support.

Comments (optional)

(Enter for a new line)

J) Are you required to backup or restore files to the NT Workstations you support?

□ No. I am not required to backup or restore files to NT Workstations.
□ Yes. I am required to have the ability to backup and restore files on the NT Workstations I support.

Comments (optional)

(Enter for a new line)

III. Domain Configuration and User Management Requirements

A) Are you required to add or remove NT Domains or establish trust relationships with other NT domains?

□ No. I am not required to add or remove domains or trust relationships.
□ Yes. I am required to add and remove domains only.
□ Yes. I am required to add and remove domains as well as add and remove trust relationships to other domains.

Comments (optional)

(Enter for a new line)

B) Are you required to add, modify, or delete **user accounts** in any of the domains you support? List the domains.

□ No. I am **not** required to add, modify, or delete user accounts at any time.
□ Yes. I am required to add, modify, or delete user accounts from time to time
□ My primary job function is to add, modify, and delete user accounts

Comments (optional)

(Enter for a new line)

C) Are you required to modify **Account Policies** for the domains you support?

□ No. I am **not** required to modify Account Policies in any domains.
□ Yes. I am required to modify Account Policies in the domains I support.
□ What are Account Policies?

Comments (optional)

(Enter for a new line)

D) Are you required to modify **User Rights** settings for the domains you support?

□ No. I am **not** required to modify User Rights in any domains.
□ Yes. I am required to modify User Rights in the domains I support.
□ What are User Rights?

Comments (optional)

(Enter for a new line)

E) Are you required to modify **NT global/local group memberships** for the domains you support?

□ No. I am **not** required to modify groups in any domains.
□ Yes. I am required to modify groups in the domains I support.
□ What are User Rights?

Comments (optional)

(Enter for a new line)

IV. Other Requirements

Appendix A – Delegation Survey

A) Are you required to have access to **special** test share-points or directories on servers to **test** applications or application components? Please list the servers, share-points or directories you are required to access.

□ No. I am not required to have access to any special share-points or directories on servers for testing purposes.
□ Yes. I am required only to restore files to user's home directories or to public directories.

Servers/Shares/Directories and/or Comments (optional)

(Enter for a new line)