At a glance:

- Defining security by obscurity
- Evaluating security by obscurity measures
- Assessing the value of renaming the Administrator account
- Making informed risk-management decisions

The great debate: security by obscurity

Jesper M. Johansson and Roger Grimes

The term "security by obscurity" is often met with derision from security people, particularly those who like to consider themselves experts. Nearly akin to a four-letter word in some circles, security by obscurity, as noted on Wikipedia (http://en.wikipedia.org/wiki/Security_through_obscurity), represents one of the truly controversial aspects of security. You will often see mocking references to people whose efforts are dismissed as "just security by obscurity."

Security by obscurity is, in a nutshell, a violation of Kerckhoffs' Principle, which holds that a system should be secure because of its design, not because the design is unknown to an adversary. The basic premise of Kerckhoffs' Principle is that secrets don't remain secret for very long.

As a very good example, take the Windows NT® LAN Manager (NTLM) authentication protocol, which was initially considered a design secret. In order to implement the Samba interoperability product for UNIX-based operating systems, the Samba team had to reverse engineer the protocol. The result was the most comprehensive documentation on NTLM available (http://monyo.com/technical/samba/translation/ntlm.en.html) as well as the discovery of a number of bugs. Because so much of security grew out of cryptography, and so many secret designs have been revealed, many security practitioners believe all of Information Security should follow Kerckhoffs' Principle.

But is security by obscurity always bad? In this article, we will explain what security by obscurity is, attempt to shed some light on why many consider it a waste of time (and others don't), and show you why the answer, as usual, is far more complicated than it seems at first.

Introduction to security by obscurity

Security by obscurity does not include measures that do absolutely nothing to mitigate a problem. For example, an organisation that blocks Network News Transfer Protocol (NNTP) at the border routers to prevent employees from reading newsgroups but permits outbound Secure Shell (SSH) is not engaging in security by obscurity. Since SSH can tunnel every protocol, the problem is not obscured; the silly mitigation that was implemented fails to do anything other than prevent legitimate users from accomplishing legitimate tasks without violating security policies. Such measures are not security by obscurity; they are just silly and pointless.

As the word "security" in the phrase "security by obscurity" implies, you do get some protection out of the measure. However, what is also implied – and this is the problem – is that you are not actually doing anything to stop an attack on one or more vectors. (An attack vector is essentially a means by which an attacker can access a system.)

Suppose you have a vulnerable Web server, for example, that can be attacked over TCP port 80 using a public exploit. To close that particular vector, you can patch the Web server or you can turn it off; either action would completely stop this vector. You could partially stop the attack vector by using a firewall or IPsec to close port 80 to all but a few select computers. This wouldn't completely block the attack vector, but it would significantly mitigate the problem.

Security by obscurity, on the other hand, involves taking some measure that does not stop the attack vector but merely conceals it. For example, you may decide to move the Web server to port 81 instead of 80 so only those who know where to find your Web server will be able to do so. Or so that argument goes. In reality, moving your Web server to port 81 stops only some attacks, and mostly just inconveniences the end user. A competent intruder will simply run a port scanner and a Web banner grabber against a large number of ports to discover Web servers on non-standard ports. As soon as he finds one, he can fire off the exploit against your server because you did not actually eliminate the attack vector, you merely (temporarily) obscured it.

Does this mean you should not even try? The answer is that it depends. As with everything else in the Information Security field, it all comes down to risk management. To understand the key factors to consider, we will take a quick look at a few more security by obscurity measures and then discuss one – renaming the Administrator account – in more detail.

> Security by obscurity involves taking some measure that does not stop the attack vector but merely conceals it

Assessing security measures

Examples of security by obscurity abound. They may be actions that system and network managers take, or they may be initiated by software developers. What they all have in common, however, is that they intend to mitigate a vulnerability by hiding it from potential attackers.

Might not some of these procedures have at least some beneficial effect? Is it truly fair to say that all security by obscurity is bad? You will certainly find proponents of at least some of these measures. For example, in Windows it is possible to hide drive letters in Windows Explorer. Many environments, most notably school computer labs, use this technique to keep users from saving data to the hard drive. Of course, most applications can still save data to the hard drive, so it provides little value as an ultimate security measure. However, the institutions that have implemented it often claim it reduces data on the hard drive.

Another type of security by obscurity practice often implemented on Windows is turning off the administrative network shares (such as C$, Admin$, and so forth). This is thought to keep an attacker from being able to connect to the computer remotely. In reality, this is not only not true, but an attacker who has an account that can use the administrative shares can remotely re-enable those very same shares. Yet many organisations swear that disabling these shares reduces the incidences of malware on their networks.

Another security by obscurity practice is turning off the administrative shares

One of our favourite examples of misplaced effort is the "Allow undock without having to log on" setting in Windows. If this is set to disabled, the "undock computer" button is hidden from the logon screen. The idea behind this is that this way an attacker cannot gracefully undock the computer. Of course, any intruder can simply stuff both the computer and the docking station into a bag and walk off with them, whether or not that button is visible. This possibility is marginally not even security by obscurity.

Another clear example is the "Allow Server Operators to schedule tasks" setting, which, exactly as it states, permits users who are members of the Server Operators group to schedule tasks. This is a sensitive issue because such tasks may run as Local System, and only administrators should be able to schedule tasks that run as Local System. Of course, server operators can potentially make themselves administrators through any number of different vectors, so preventing them from scheduling tasks in this manner actually has minimal value.

Yet many organisations like this setting because it allows their engineers to be server operators instead of administrators, which means they are less likely to accidentally destroy the server. That, in and of itself, may carry some benefit.

So what's the verdict? Obviously, some of these issues can be very complicated. We have spent many enjoyable hours debating these types of measures. Roger is firmly in

Don't change defaults by Steve Riley

I'm on the "don't change" side of the renaming issue. This really isn't even a security matter – it's a systems management issue. And as I spend more time thinking in the systems management space (because management and security are becoming one), I become more and more a fan of not doing anything that can potentially increase a system's brittleness. One sure-fire way to increase brittleness is to change things from defaults. There are two (well, OK, three) reasons people change default settings:

1. There is a known requirement for the functionality the change provides.
2. There is an assumption that the change will improve security.
3. (Alert: silliness ahead.) Someone read about it in a magazine article.

If you need to change from a default name because of reason number 1, go right ahead and do so. These kinds of changes rarely result in system instability, often because they have been previously tested.

If you are making a change from a default for reason number 2, please stop first and reconsider. These kinds of changes are almost never tested by the software manufacturer, and therefore the manufacturer can't predict how the system will behave after you've made the change. Furthermore, there are generally much better alternatives that will provide you with true security.

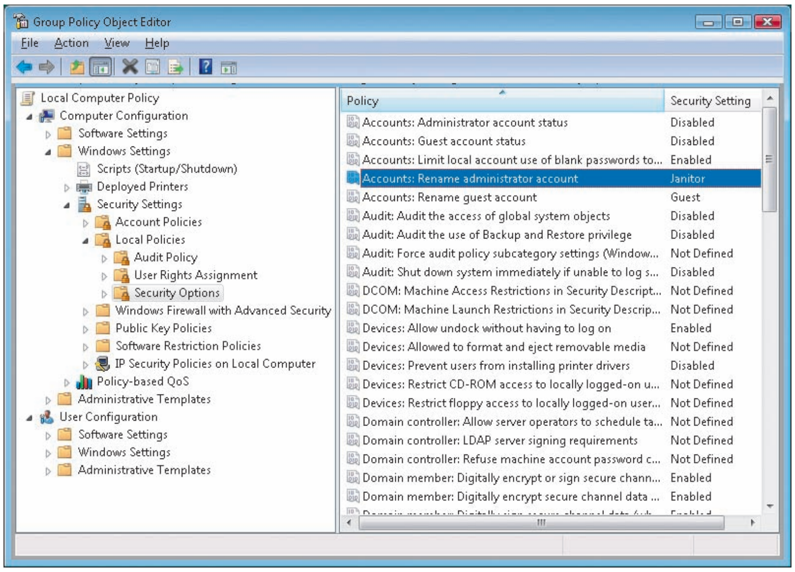
If a bad guy happens to know an account name, so what? It's the password, and the strength of that password, that keeps the bad guy out of your system. The urge to change default account names like Administrator and Guest to something not so easy to guess is often actually indicative of a desire to avoid strong passwords or passphrases. Fix the real problem (lousy secrets) and you can then avoid brittleness (changing default names).

the camp that finds value in such practices. Jesper, on the other hand, believes they are, at best, a waste of time in most cases. Let's explore one of the most often cited – and disputed – cases for security by obscurity: renaming the Administrator account. Roger will argue for the measure, Jesper against. Security notables Aaron Margosis and Steve Riley also weigh in, in the sidebars "Rename the Administrator Account" and "Don't Change Defaults," respectively.

Hiding the administrator account

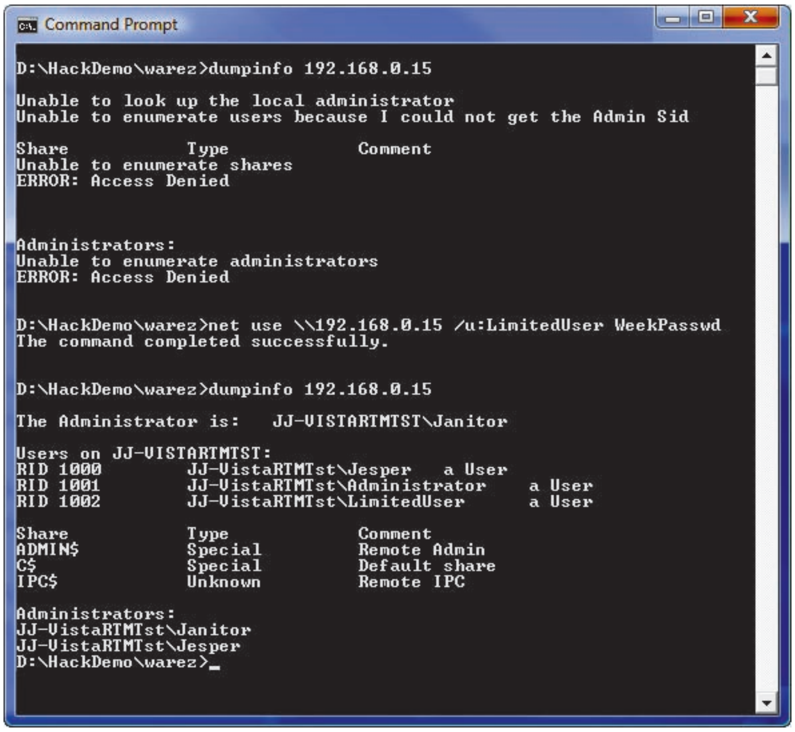
Renaming the Administrator account, relative identifier (RID) 500, to something unknown by the general public is often recommended by security professionals and is included in several Microsoft hardening guides. Group Policy even includes a setting for renaming the Administrator account, as shown in **Figure 1.** The idea is that if the Administrator account is renamed, an attacker will have a more difficult time logging in as the true administrator. Of course, the obvious problem with using Group Policy for this operation is that the renamed Administrator account will have the same name on every computer to which this Group Policy object was applied. That lessens the obscurity value of this particular security measure.

It is also important in this context to realise that any user with a legitimate account can retrieve the name of the Administrator account, regardless of whether it has been renamed or not. The Administrator account always has a RID of 500. By simply asking for the name of the account with RID 500, any user with an account can see what it is really called, as shown in **Figure 2**.



Policy	Security Setting
Accounts: Administrator account status	Disabled
Accounts: Guest account status	Disabled
Accounts: Limit local account use of blank passwords to...	Enabled
Accounts: Rename administrator account	Janitor
Accounts: Rename guest account	Guest
Audit: Audit the access of global system objects	Disabled
Audit: Audit the use of Backup and Restore privilege	Disabled
Audit: Force audit policy subcategory settings (Window...	Not Defined
Audit: Shut down system immediately if unable to log s...	Disabled
DCOM: Machine Access Restrictions in Security Descript...	Not Defined
DCOM: Machine Launch Restrictions in Security Descrip...	Not Defined
Devices: Allow undock without having to log on	Enabled
Devices: Allowed to format and eject removable media	Not Defined
Devices: Prevent users from installing printer drivers	Disabled
Devices: Restrict CD-ROM access to locally logged-on u...	Not Defined
Devices: Restrict floppy access to locally logged-on user...	Not Defined
Domain controller: Allow server operators to schedule ta...	Not Defined
Domain controller: LDAP server signing requirements	Not Defined
Domain controller: Refuse machine account password c...	Not Defined
Domain member: Digitally encrypt or sign secure chann...	Enabled
Domain member: Digitally encrypt secure channel data ...	Enabled

Figure 1 **Renaming the Administrator account**



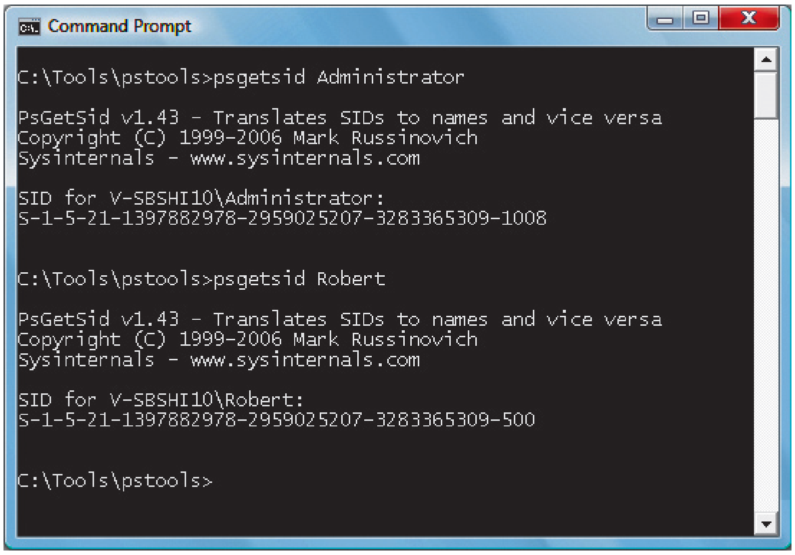
```
D:\HackDemo\warez>dumpinfo 192.168.0.15

Unable to look up the local administrator
Unable to enumerate users because I could not get the Admin Sid

Share           Type            Comment
Unable to enumerate shares
ERROR: Access Denied


Administrators:
Unable to enumerate administrators
ERROR: Access Denied

D:\HackDemo\warez>net use \\192.168.0.15 /u:LimitedUser WeekPasswd
The command completed successfully.

D:\HackDemo\warez>dumpinfo 192.168.0.15

The Administrator is:   JJ-VISTARTMTST\Janitor

Users on JJ-VISTARTMTST:
RID 1000        JJ-VistaRTMTst\Jesper   a User
RID 1001        JJ-VistaRTMTst\Administrator    a User
RID 1002        JJ-VistaRTMTst\LimitedUser      a User

Share           Type            Comment
ADMIN$          Special         Remote Admin
C$              Special         Default share
IPC$            Unknown         Remote IPC

Administrators:
JJ-VistaRTMTst\Janitor
JJ-VistaRTMTst\Jesper
D:\HackDemo\warez>_
```

Figure 2 **Finding renamed Administrator accounts**

Roger's take

The main argument I hear against renaming the Administrator account is that it is so easy to convert any security principal account name into its related security identifier (SID) and find out its RID. And the true Administrator account always has a RID of 500. So, if an attacker can easily convert user account names into the SID/RID combinations and find RID 500, why bother?

But it's not so simple. In order to accomplish user account name-to-SID/RID translations, you must have access to either the NetBIOS or LDAP ports. Most perimeter fire walls don't allow that type of access over the Internet, so until the attacker completely bypasses the firewall, he won't be translating anything. Furthermore, anonymous SID translations don't work against Windows XP and later versions of Windows, except on domain controllers (DCs). Against most externally facing computers (the ones most at risk), the attacker would need authenticated



```
C:\Tools\pstools>psgetsid Administrator

PsGetSid v1.43 - Translates SIDs to names and vice versa
Copyright (C) 1999-2006 Mark Russinovich
Sysinternals - www.sysinternals.com

SID for V-SBSHI10\Administrator:
S-1-5-21-1397882978-2959025207-3283365309-1008


C:\Tools\pstools>psgetsid Robert

PsGetSid v1.43 - Translates SIDs to names and vice versa
Copyright (C) 1999-2006 Mark Russinovich
Sysinternals - www.sysinternals.com

SID for V-SBSHI10\Robert:
S-1-5-21-1397882978-2959025207-3283365309-500


C:\Tools\pstools>
```

Figure 3 **Log in attempts to a decoy account named Administrator can serve as an early warning**

credentials to do name-to-SID translations.

So there are a fair number of real defence-in-depth hurdles that have to be bypassed in order to begin a translation attack. And even if the attacker gets this far, he ends up in the same spot he'd be in if the account name wasn't renamed. Renaming the Administrator account can only improve security. It certainly can't hurt it.

It's another secret If an attacker does not know the Administrator account name, it becomes yet another "secret" label, analogous to a password, which an attacker has to know. User account names are significantly less likely to be as complex as an administrative password, but it's still one more hurdle that exponentially complicates the problem of password guessing/cracking. If you've ever done password penetration testing, you already know how much more work it is to have to guess both the user name and the password. It compounds an already difficult task and makes it that much more difficult.

If you rename your Administrator account, you immediately kill all malware that relies on the name

Thwarts automated malware and script kiddies I've been practising Windows security defence for 22 years now, and I've been running eight Windows honeypots, exposed to the Internet, for the past 5 years. In all of that time, I've never seen automated malware (which comprises the vast majority of all attacks) ever use any user account label except Administrator (or root, in the case of *NIX systems). If you rename your Administrator account, you immediately kill all malware that relies on the administrator name. And that translates to decreased security risk.

The same goes for script kiddies. Every "kewl" Windows hacking manual I know mentions name-to-SID translation techniques, but for some reason, I've never seen a case of it on my honeypots if I've also had a "fake" Administrator account to throw them off. Good hackers should always verify that the Administrator account they found is the real Administrator (RID 500), but they don't. I don't know why. I blame it on lazy hackers and ordinary human behaviour.

Stops most professionals, too This surprises most people. When you've been running honeypots for a few years, you quickly recognise the difference between automated attacks, script kiddies and professionals. In the past five years, with millions of reported attacks against my honeypots, I've never seen the professionals do SID translation when the fake Administrator account was present.

I'm sure some of the professional criminals do undertake SID translation, but I'm willing to bet that's a small minority, and one I've never documented in the wild. Why wouldn't the professionals do it? I'm guessing that they don't see any reason to try something when the majority of the world isn't doing it.

Simplifies alerting Now, the other side may contend that if renaming the Administrator account became prevalent, it would lose its value as an obscurity technique. The argument is that malware, script kiddies and professionals would change their default tactics to look for names other than just Administrator. And this is a valid concern. Fortunately, it doesn't change the essential condition. First, if the Windows super-privileged account isn't named Administrator, the malicious hacker must work harder. That's just a fact. And if the malicious hacker has to work harder, he is a little less likely to try that avenue of attack, perhaps allowing one of the other offsetting defence-in-depth techniques to detect the malicious activity faster.

And that leads me to my last point in favour of renaming. If your Administrator account is never called Administrator, and you create another account with that name, as shown in **Figure 3**, you have a good alerting mechanism. If your event monitoring detects an attempted logon using the default name, it is an event to be immediately explored.

Our event logs are full of "bad" logons that don't mean anything. Typically, it's just a user (or Windows) trying to log on using a bad password or something like that. If your Administrator account is called Administrator, though, how can you easily separate good from malicious logon attempts? If you never have a logon account called Administrator, and you detect an attempted logon using that account name, you know the intent is probably malicious. Early warning with low noise has much value in today's defence.

Jesper's turn

As with the arguments for, there are valid arguments against renaming the Administrator account. Before getting into that, however, I must concede that Roger's last point is quite valid. However, in a highly managed environment, *any* logon with the Administrator account should be investigated because that account should never be used other than in emergency recovery situations.

It's superfluous The main risk that renaming the Administrator account is supposed to mitigate is that of someone guessing its password remotely. But only a user who does not have another authorised account on the computer would be thwarted by renaming the Administrator account. Such an attacker would typically try a series of random passwords to log into the Administrator account. However, they'd receive the same error message regardless of whether he guessed the account name wrong or the password wrong.

That suggests that one of the primary arguments for renaming the Administrator account – that the script kiddies go away – is flawed. They do not go away any sooner when the Administrator account is renamed than when it has the original name because they can't tell! They will guess the same set of random passwords no matter what and then move on to the next potential victim.

This means that as long as the password on the Administrator account is strong enough to repel attacks, renaming the account provides no additional value. Let us say we have a 15-character password on our Administrator account, composed of upper- and lowercase letters, numbers, and symbols chosen from the entire keyboard. Guessing that password across the network will take roughly 591,310,404,907 years. By comparison, that is approximately 43 times longer than the universe has existed.

As long as the password is strong enough to repel attacks, renaming the account provides no additional value

Now, let's say we rename the Administrator account, and we make it one of 1,000 possible values. We would extend the time to guess the password to 591,310,404,906,787 years, or 43,161 times longer than the universe has existed. Are we more secure? Sure, we are three orders of magnitude more secure. Did we make it any less likely that the attacker will guess our password? Well, it is infinitesimally small in either case. In other words, in real terms, we are absolutely no better off than with the original Administrator account name. Just because renaming the account kills malware that attempts to use it does not mean that malware would have been successful had you not renamed the account. In fact, if you use a strong password, as you should, you can virtually guarantee that it would not have been successful, regardless of the name of the account.

It is clear that much security guidance requires renaming the Administrator account; but that does not make it a valuable, or even valid, security measure. It simply removes your ability to make a proper risk management decision about it. Security guides have very frequently required settings that make no difference, and in many cases, settings that do not even exist. Eventually, to progress in the field of security, we must step above the requirements and actually analyse the issues, and assess whether the mitigations make sense or not. It is important to remember one critical point here – the fact that an attacker is targeting a feature is, in and of itself, not a sufficient reason to modify that feature. You should modify a feature only if you have a reasonable expectation that an attack will be successful unless you do modify the feature.

If we assume a strong password, the likelihood of success is effectively zero whether the account is renamed or not. Therefore, as long as you have a strong password, you have no particular security reason to rename the account. Moreover, if you, like me, operate on the principle that a computer will probably be more stable the fewer tweaks and changes you make to it (a well-established fact, by the way), that's even more reason not to rename the Administrator account.

Shifts focus to low-value mitigations One problem with low-value security by obscurity mitigations is that they risk shifting the organisation's focus from higher-value mitigations. For example, the time and effort put into renaming the Administrator account can't be used on something else. If that something else provides higher value than renaming the Administrator accounts, the organisation has lost an opportunity. This may not sound like a real cost, but imagine renaming 50,000 Administrator accounts, and you start to get the idea.

Worse still is the very real possibility that the organisation's leadership will rest once

Rename the administrator account by Aaron Margosis

Jesper, in an ideal world you would be absolutely correct. Passwords would always be strong enough to make brute-force guessing unfeasible, and -500 local admin accounts would only ever be used for emergency recovery. In the real world, though, neither is true.

Despite the great security evangelising you have done, and in particular the terrific passphrase series you authored ("The Great Debates: Pass Phrases vs. Passwords," Parts 1, 2 and 3, at www.microsoft.com/uk/greatdebates1; www.microsoft.com/uk/greatdebates2; and www.microsoft.com/uk/greatdebates3), many sysadmins do not have an up-to-date understanding of what constitutes a strong password.

It wasn't very long ago that a password made up of eight random characters drawn from multiple character sets was strong; Moore's Law rendered that obsolete. User (and sysadmin) education is a weak link and most likely will remain so, at least until the topic of password strength becomes a hot item for the cable news channels to obsess about. So, given that real-world password guessing today doesn't require 600 billion years and can instead often be done during lunch, adding a x1000 multiplier can have real value. And against the many automated attacks that try to pound on the account named Administrator, renaming the account renders it invulnerable.

The time and effort typically involved in renaming the admin account is usually low; typically, as you noted, it's a simple GPO setting. The U.S. Government's Federal Desktop Core Configuration (http://nvd.nist.gov/fdcc), in fact, requires renaming the -500 account. The rename is just one of many required settings and not one I've seen taking an inordinate amount of time or attention. Nor have I seen anyone lulled into a false sense of security regarding its value – I have yet to hear anyone say, "We don't need patch management because we renamed our admin accounts."

Does the rename have any value when the account is renamed the same across an entire organisation? It's not a huge value, but there is some: for one, it blocks automated attacks that expect Administrator, and second, the set of potential attackers is not necessarily a subset of the "everyone" who knows the new name. (The bigger risk is when local admin accounts – renamed or not – share a common password across an organisation. Managing those remains the bigger and more important problem. Disabling the -500 account is one way to address it, but this blocks an important recovery avenue when domain accounts can't be used.) I'll also point out that our security guides have long recommended renaming the default accounts, so doing so has been well tested and is fully supported.

I think we all know by now that obscurity measures—*by themselves*—do not constitute adequate defence. But they can enhance other security measures, and Roger's data shows that quite clearly. As long as the cost of implementation is low, organisations should not rule them out.

There are pros and cons to any scheme. Do the risks outweigh the costs?

the low-value mitigation is in place. Management may not always understand the minimal value provided by security by obscurity mitigations and may not take additional steps. This can actually pose a significant risk to the organisation, though the risk is quite avoidable if management simply devotes time and effort to understanding the value of the mitigations being implemented.

Expensive to manage Finally, depending on how well the mitigation is implemented, it can become quite expensive to manage. If you simply set a Group Policy Object (GPO) to rename the Administrator account, the cost is very low. Everyone will know the name, and the cost of deployment is nearly nil. However, the value it provides is also quite minimal because, as I said, everyone with an account will know what the name is. Thus, to really get much value out of this mitigation, you need to use different names on different hosts, and that becomes quite an expensive system to manage.

You would be better using a tool like passgen (www.protectyourwindowsnetwork.com/tools.htm) to set a 100-character completely random password on all the Administrator accounts across the network and use different accounts for day-to-day management instead. Considering that the Administrator account is purely for disaster recovery purposes (except on Windows Small Business Server 2003), this should not impact your network management system at all. Furthermore, the attacker would have a better chance of finding a needle on the bottom of the Pacific Ocean than correctly guessing the password on any of your Administrator accounts. Focus instead on strong, unique passwords and the script kiddies can keep guessing all they want.

It's all about risk management

Virtually any security by obscurity measure can be analysed the way we have here. There are pros and cons to any scheme, and the right approach for one organisation does not necessarily fit a different organisation. In the end, this becomes a risk management problem. Do the risks outweigh the costs of implementing the solution? Every decision you make in protecting your information assets must be an informed risk management decision, and choices are rarely black or white.

True, some decisions have already been made for you. For example, while you can certainly choose not to encrypt credit card information or store the credit card verification codes, either action will likely result in a loss of your ability to accept credit cards as a form of payment. The probable negative impact to your business of that decision is such that you do not really have a choice. In other words, while this is most certainly a risk management decision, it is one that is quite easy to make.

Likewise, it's also true that nobody in his right mind would connect an open wireless network to the back end of the in-store network in a physical store. However, that doesn't mean that the decision is not a risk management decision. It is. One may chose to do so, and if one does, one should suffer the consequences (which never seems to actually happen, unfortunately).

The key step to take here is to clearly articulate the problem you need to address, the proposed solutions, and the pros and cons of each. Once you have done that, you can make an informed risk management decision. Without that information, you are making decisions based on a hunch, and those are rarely good decisions. ■

JESPER M. JOHANSSON *is a Software Architect working on security software and is a contributing editor to* TechNet Magazine. *He holds a PhD in Management Information Systems, has more than 20 years of experience in security, and is a Microsoft MVP in Enterprise Security. Jesper's latest book is the* Windows Server 2008 Security Resource Kit.

ROGER GRIMES *(CPA, CISSP, MCSE: Security), Senior Security Consultant for the Microsoft ACE team, has authored or contributed to eight books on computer security and authored more than 200 magazine articles. He is a frequent security conference speaker and the* InfoWorld *security columnist.*