Introduction to:

Basic Security Concepts

A Guide for Administrators and Home Users on the design and implementation of security for your network.

By: Robert H. Williams III
CompTIA A+\Net+, Microsoft MCP\MCSA\MCSE (2000), Cisco CCNA

Version 0.9.7.1.4

Document Contents:

Introduction

About this paper:

This document is about the basics of information security, covering both design and implementation topics. It shall be covering system and network security, a more in depth look into Windows security, and will be covering some security Design issues. Some of the information contained here-in is about abstract concepts in security design. Other bits of information are about actual configuration steps that can be used to lock down Microsoft Windows. But to understand what needs to be configured, you should first look at why it needs configured, and how it needs to be protected, and this is what most of this paper is about.

To keep this paper easy to read for the majority of home users, you may wish to skip to the *What home users should know about Information Security* section below.

The level of knowledge required to use this document is low. While some items covered within may apply to certifications such as Security+, MCSE:Security, and even CISSP, this should not be considered a study guide for any of these certifications. It's more of a tie-in, to help the reader grasp the relationship of all the information presented for these levels, and an introduction for home users to higher levels of security. This exams come at security from different standpoints. For example the MCSE:Security doesn't stress data backups as much as the Security+ exam, because it's assumed the MCSE student knows about them already (Backups are covered in the core exams, and also in the Security+, and the Security+ exam is one of the two security electives the MCSE:S candidate may take). Sections designed for home users will be marked, as mentioned below.

Introduction:

Security is a hot topic for debate. Just look at who talks about it, from Congress and your company's local IT department, to Internet forums and even other home users, even purple monkeys on the Internet trying to sell you firewalls and be your "buddy", you just can't seem to escape the topic of security. With areas ranging from SOX and HIPAA, down to the latest browser worm, there's a lot of ground to cover, and much more ground to cover if you wish to understand it. And because most media outlets just do not understand what security is about, perceptions on security are skewered. This paper is designed to help bring some security concepts to light. It's designed to teach the user not to react to known threats so much as preparing for unknown threats.

Prerequisites:

Most of this paper's implementation section is about Microsoft Windows, so you should use a Windows based machine for that part. And of these configuration, most applies only to the NT line of Windows systems, NT, 2000, XP, and 2K3, and Vista. And of course you will need to be the administrator on said machine for most of these changes to take place.

Because this paper is about security design and implementation, and risk mitigation, it can be performed on any OS. Much of the paper covers network design in general, and will never touch on the underlying OS.

To get into some of the more advanced items of security, an Active Directory domain is recommended. Basic instructions on getting one up and running will be talked about in the Windows area of the Implementation section.

What home users should know about Information Security

Most of the information contained within this paper is aimed at IT professionals, such as network and system administrators. While it's presented in a manor that most people can understand, it can still be quite daunting for most home users. To help home users get what they need from the paper quickly, simpler sections covering the basics are marked with **HOME**. This mark will show the simpler overviews of the most important information, and cover basic steps needed for security that do not require much background knowledge.

Design Topics

What is IT Security about?

As most people know, Information Technology (IT) Security is all about information and protecting it. However, to protect the data, you need to know what is type of protection it needs. Protecting data on your publicly viewed web site isn't the same as protecting your payroll data. One, you want people to see it, and the other, only a few people can see it. This is where the data types come into play, and will be the first item that needs covered.

What are the four data types?

There's four types of data information can fall under, Public, Internal, Confidential, and Secret. The damage is done just by exposing this information in many cases, while in others, it is only damaging if the data is changed or destroyed.

Public:

Public data is designed to be shown, so there is no reason to protect it from being seen. If Public data is changed or destroyed, however, you lose something you can remember by the letters PTR: Prestige, Trust, and Revenue. Public data needs to be accessible, but only a few users or machines should be able to change it. Examples of Public data for businesses may be information on your company web site (this does not include any configuration files used by the website however), or on documentation sent to consumers. For home users it may be your average email or instant messaging data. While it would do no harm for this data to be seen by others, if this data was changed in transit, the results could be disastrous.

Internal (Private):

Internal data, also called Private, is data company workers generally know, but outsiders don't. It's items such as PINs (Personal Identification Numbers) for doors, internal procedures, or the like. It's information that most company workers can find out. Discovering this information is normally not a risk in itself, but it allows for better attacks. The main risk is modification, either by an outside force such as an attacker, or most cases, accidentally by an internal user. This will generally affect the operations of a business, and not much else. Most files on your OS would actually fall under this, and damage to them will only affect operations. Keep in mind, this internal data can also be a stepping stone to launch attacks on other, more secure, forms of data. Removing internal data from the view of workers can cause damages to business Operations, performing a Denial of Service (DoS) in extreme cases. For a home user, Private data could be where you store your keys, security codes for home security systems, to even less obvious items. While knowing your pet's name may not seem like any sort of security risk, if you happened to use your pets name as a prompt in case you forgot your password, this could raise some security concerns.

Confidential:

This is the data used by a limited number of internal users, and should not be known to the majority of workers. This is the class Human Resources (HR) data and Payroll Information falls under. Read access to this data is limited to a few users, and write access is generally restricted even more. If this becomes public internally, Operations and Internal Trusts are at stake, while if reviled externally, you lose your PTR, along with Operations and Internal Trusts. OS files dealing with security also fall into this area in most cases. Confidential data is just a few steps away from Secret data, and like Secret, it needs to be protected. For a home user this could be some emails you've wrote, browser history, or a folder containing pictures and movies the rest of the household wouldn't approve of.

Secret:

This is the data most people think of when they hear about breaches in information. This data is your trade secrets, intellectual property, and External Secrets, such as info held in trust for others (Partner company's, or customers). Loss of this data may cause critical damage to the company, and could very well be the downfall of it. Besides the PTR loss, and maybe loss of Operations, there's fines and legal actions to think of in most cases.

While this may seem like only businesses would have data that fall in these four classes, all information can be placed inside them, sometimes into more then one class. As stated before, most of the files used by your operating system would fall under Internal data. It's not something that needs to be kept secret so much as needs to be kept from being changed. Music files on your machine? They have an effect on the Operation of how you run your life, and so fall under operations. Credit Card info could be considered Secret.

How is the security of the average company setup?

Many businesses, and almost all home users, have a security model that resembles an egg shell. While the outer surface is hardened to keep people out, once there's a breach, there's nothing internal to prevent or limit access. And unlike the mighty chicken, companies have no way to make this security eggshell without holes in it. If a computer is online and networked, then any flaws in the firewall, along with any programs that access the network, form the security holes. If the computer is not online, physical access is still the hole in the shell. If it's a server in a locked room, the door itself is the hole in the shell. Before their move to make Windows more secure, Microsoft applied to get a security certification level of EAL-4 for Windows 2000. And while it did receive EAL-4 status, the only configuration that received this level was a non-networked machine. And if you're wondering, EAL is on a scale of 1 to 7, and while 4 is a great ranking to earn for a mass-produced commercial OS, the fact that it only received a 4 out of 7 should show you that there's more to security than meets the eye. More about EAL levels and exactly what they mean, along with rankings of other products, are in the appendix.

The area of the shell you can attack is known as the Attack Surface. As you can imagine, a smaller Attack Surface is better, but this is only a tiny part of security. Most people and many companies think that the attack surface is the most important part of security, or even that it's all there is to security. This is putting all your egg-shelled networks in one basket. While it may be hard to get into the network, once in, everything is at the mercy of the attacker. It only takes one hole in the shell to get in, and as stated before, you can not prevent all holes in a shell.

The primary goal in IT security is to limit, not prevent, damage. While preventing damage is great, you can never prevent 100% of damage. Rather, you must try to make it harder to cause the damage, and work on lowering how often that damage can happen. While this concept is covered more in-depth later, a quick overview is as follows: all damage done can be assigned a monetary value. If your home computer was completely destroyed by fire, how much money would the damage be? While there is the cost of the computer itself, there is also the data inside the computer, damaged. Even if you recovered the physical cost of the computer, the data contained within could have more value of the computer itself. However, if you had an up-to-date backup of the data that was not destroyed in the fire, then the damage done would be the cost of a new computer, and effort required to restore said backups While you have not lowered how often the event may happen, you have lowered the damage done by it.

The egg-shell rule, while it applies to most companies, applies to home users too. Think for a moment. Chances are, you're browsing the Internet right now, from the internal servers on your home or work network, ones with secret level data, as an administrator! And most likely you also have programs running as servers, such as yahoo messenger, with full administrative rights, and no damage control.

Now at this point I must state this clearly: While having a strong shell is not only desirable, but almost required for security, other security steps must also be taken. I am not saying you do not need a "shell" around your system, and I am defiantly not saying you do not need a firewall. I'm saying that this is only the beginning to security, and that other steps must be taken. These include secondary shells under the first, access controls, not having systems trust other systems (Or even allowing one subsystem to trust another sub system). And perhaps one of the biggest issues with companies, making sure security procedures are followed. While you may have designed a great security setup for all your machines, are your administrators in remote offices really following them? Do the administrators in your data room know how to react in the event of a crisis? If you walked into your server room, and just unplugged a server at random, what would happen? How quickly would your network recover? This is something you need to start thinking about. I've never met a firewall yet that would plug a server back into the wall.

How do most security breaches happen?

If you were expecting to see the word "hackers" then you've been watching too much television. Most security related issues are accidental issues. Blame does not always fall on a person, mind you. If the operating system crashes and is unrecoverable, chances are it was caused by a hardware error, but could have been caused from something as simple as installing a bad bit of software. While finding the true cause will help you prevent it from happening again, tossing blame around at anything you think caused it will not. If it's not an object that you can control, then blaming it is no good. If a hurricane wipes out your data center, do you blame the hurricane? No. But you can place blame on the location being in Florida, and to prevent this, you could move the base of operations to a new location. The problem with this, you may open it up to tornadoes or worse.

The most common security breech you hear about is cause by automated attacks, such as Worms, Viruses, and Trojans. Because these are automated attacks, they can take out a lot of machines very quickly, but they also can not adapt beyond what they were programmed to do, and can not compensate to defenses they were unprepared for. For example, if a worm was to try to log in as administrator, and you renamed the account, it would fail. However, any attacker worth his salt would be able to see the administrator account was renamed with ease. And a worm can be programed to check to see who the administrators of a machine are. When implementing security, you should keep in mind, is the effort worth the gain? And will this effort make it harder to use the system, thereby raising the operational cost?

Also keep in mind, damage is not always caused by external sources. Internally caused damage is a major issue. In terms of companies, most security breaches are internal, caused by users. This applies to things you may not take into consideration. It could be the VP of marketing meant to send an email to a friend, talking about how cute the new female hire is, only to accidentally send it to the whole office she works in. It could be that the newest security patch for X program accidentally changing some system files, preventing the OS from booting. Even hardware failures, from power outages, to blown CPUs, fall under security due to the disruptions of operations. So now you should look at what security really is.

The security triad: the other CIA.

When talking about information security, there's three things to keep in mind: Confidentiality, Integrity, and Availability, or CIA. Confidentiality comes into play when you do not wish for data to be known, and is generally controlled by encryption, Access Control Lists (ACLs), Firewalls, and the like. It applies to all data types other than Public data. Integrity is making sure the data has not been altered or damaged. Signing of the data is the main way this checked, but access controls, and other security conventions, are required to prevent the change of data. Integrity is needed on all data types. Availability means that data must be up when needed, and can be referenced when required. Backups of data are the biggest thing here, but access control and redundant hardware are also major players here. Availability may not be as big of a requirement on some data than others, and the level of availability required should be decided on a case-by-case basis.

Design vs. Implementation Example

Note: This section is lengthy, and does not cover any topics of major importance. It is here just to help show you how poor design can be overcame with good implementation, and good design can be brought down by bad implementation. And a good design can help limit the damage done by a bad implementation.

When talking security, or many other issues in the IT field, Design vs. Implementation comes into play. Design is how well it's thought out, while Implementation is how you actually do it. When talking about Design vs. Implementation, one source that's always nice to read is the Tanenbaum-Torvalds Debate. Linus Torvalds, creator of the Linux operating system, and Andy Tanenbaum, creator of the Minix operating system, had a debate in some newsgroups that can help show you the basic design philosophy of Linux and how it differs from the Unix it was designed to be like.. Linux's basic kernel design is that of a monolithic kernel, where there is a lot of unneeded functions stuffed inside the kernel. Minix is a micro kernel, where only the most needed of functions are in the kernel. The debate is interesting, because while Tanenbaum was preaching about the virtues of a micro kernel, Torvalds agreed with him. So why did Torvalds implement a monolithic kernel? Ease of implementation. While the micro was a better design, Linus felt it was too hard to correctly develop for, and as such, Linux would never have been a completed operating system. While a monolithic kernel does have inherent design issues, it's simpler to make work arounds for these design issues than in a micro kernel. No one will say a monolithic kernel is more secure then a micro kernel of the same level of completeness and of the same quality level of design, but in general the micro kernel took much longer to develop to reach the same level of quality as it's monolithic brother. The implementation can help overcome flaws in the basic design.

This should also let you see some issues. While it was simpler to develop, the monolithic design of Linux has limited it in many regards for security. Like with most consumer level OS's such as Unix and Windows, security was an after thought of Linux, and added in later in the development. The design of it isn't as secure as some other operating systems designed for security, such as Green Hill's Integrity operating system. But the security of each operating system's implementation is up to debate. Is a micro kernel with 1 security issue that allows administrative access more or less secure then a monolithic kernel with no issues? And what happens when the micro kernel's 1 security issue is now patched? Or, to put it in perspective, say operating system A has 2,000 known issues in it's lifetime, all stemming from about 80 fundamental design flaws. System B is lesser known, and has about 450 known issues, but they stem from over 200 separate design flaws. Which is more secure?

Well based on that information, and that information only, they are the same. What's easier to enter, a house with one door, or five? Well since you can only enter one door at a time, they're the same. But like all things, this isn't quite the whole truth. In operating systems, say you had 50 holes, while the other operating system had 20. Well, that's 50 items you have to block, vs. 20. But say all 50 holes can be blocked via a third-party firewall, while 1 of the other operating system's 20 issues can not be blocked via firewall, because it's a service that requires Internet access. While the firewall isn't part of the OS, it allows it to block all the holes in the OS, while the other OS can not block all the holes.

One hole, or fifty, it doesn't matter, it's still a way in. Rather then plugging holes in your egg shell, then waiting for new ones to be discovered, holes that were always there, and then plugging up these new holes again, you need to set up your defenses farther in, on the items needing protected themselves, and then work outward, blocking all you can. The goal is to make it so no holes "line up", meaning what can get past one line of defense will not get past the next. Say you have two firewalls in your DMZ, both Cisco PIX, and an attacker knows a way to bypass the PIX firewalls. Well if he can bypass the first, chances are he can bypass the second. By swapping the second PIX with another vendor, such as a ISA 2004 appliance, you shift where your "holes" are located. Now they need a way past both the PIX and ISA. And if there is a new issue with your ISA server, the PIX will block it. If there's an issue with PIX, the ISA will stop it. Keep in mind though, if you have any sort of web access, you've poked your own holes in both of these "shells", holes that lead right past both firewalls into your network.

Introduction to Risk Assessment.

Note: This is a very basic coverage of one type of risk assessment. It's not this simple, but this will give you the idea of how basic risk assessment works. There are other types of risk that can not be used on this scale. For example, risk that comes tied with potential gain, such as a new business venture, does not mesh well with this formula. It also doesn't cover how to figure out the ARO or SLE values.

Risk Assessment is the practice of identifying risks to your business or personal assets, assessing the potential damage done, and recording how often said risk will occur. In Risk Assessment it's not if an event will happen, it's how often. It is only after you know the value of each item that you should begin to cover your assets.

In Risk Assessment there are three principle items to remember: Single Loss Expectancy, Annualized Rate Of Occurrence, and Annualized Loss Expectancy.

Single Loss Expectancy (SLE) – This is the value of damage the average event of this type will happen. For example, if the average downtime of a server would cause $1,000 in loss per hour, then your SLE for this event is $1,000.

Annualized Rate Of Occurrence (ARO) – This is how often per year an event will happen. If your server goes down once for 4 hours per month, then you have an ARO of 48. If it went down for 1 hour every 4 years, then your ARO for this event is .25.

Annualized Loss Expectancy (ALE) – This is how much, per year, you are losing from this event. The ALE is generated by the SLE x ARO to generate your ALE. For example, with the SLE of $1,000, and an ARO of 12, your ALE is $12,000. If the ARO was .25, then the ALE would be $250.

Also note that these values are almost never as simple as this. If server downtime is for extended periods of time, the SLE may actually raise. For example, many company have the terms level 1, level 2, and level 3 downtimes. If a level 2 downtime happens, you begin calling in more people, people who may be on overtime. And on a level 3 downtime, it may be standard policy to call the manufacturer of the server, for instance. This causes a staggered rate for your SLE and can complicate matters. Because you have higher SLEs after longer outages, reducing the time of your average outage will pay off more then it would for other companies.

If your server had a 99.9% uptime in a 24/7, 365 day operation, then out of 8,760 hours in the year, it was down for about 8.768 hours. This translates into $1,000 x 8.768 = $8,768 average loss due to downtime. The way your average companies goes about trying to lower this would be to raise uptime. If the uptime was brought up to 99.999%, downtime is 0.0876 hours, or about 5 minutes per year. This would result in the ALE of $1,000 x .0876 = $87.60. Sounds good, right?

To get 99.999% uptime, or a downtime of 5 minutes per year, you would need a pretty hefty expense account. Chances are, your power company doesn't average less than 5 minutes of downtime per year to your location. One 15 minute blackout will ruin that ratio for the next three years. A good UPS can only last short periods of time powering a high-powered server, and backup generators can cost a bundle. And blackouts are just one item out of your control that can cause downtime. This magical 5 nines of uptime is not an easy goal to reach year after year. To go from a 99.9% uptime to 99.999%, chances are very good that it will cost a lot. How much? Depends on a lot of items, but chances are, more then your $8,768 per year lost due to down time. Is working on the uptime worth it? Might be, but remember, there's two parts to every ALE.

We've mentioned security not being about just preventing something from happening, but limiting the damage done when it does happen. Now you can see the math. Say your server has a SLE of not $1,000, but $6,000 per hour. And you averaged 9 hours of down time last year, so your ARO is a nice 9. Your ALE for the year was $6,000 x 9 = $54,000. It was estimated that the cost to cut the downtime in half, to 4.5 hours per year, would be in the range of $250,000. By lowering the downtime, your ALE would drop to $6,000 x 4.5 = $27,000, and would save your company a like amount, 27,000 per year. With this rate, the changes to the uptime would start to pay off by the 10th year, beyond the expected lifetime of the current setup. Next you look at lowering damage caused per outage. If you can lower the damage done per hour from $6,000 to $4,000, for example by using a less powerful server as a fall-over backup, then the ALE would become $4,000 x 9 = $36,000, a savings of $18,000 per year. I've seen a few companies, when upgrading a server, keep their current server as a backup, and sell their backup server. Then the next time they upgrade, their current server becomes the backup, and once again they sell the backup. This may not be piratical for most companies, especially with cost of floor space, but it is something to keep in mind.

So what's simpler to do, lower the SLE or the ARO? Depends on the events in question. If you're in the banking industry, and are processing information from external company, uptime is a requirement to begin with, and you may very well be forced to strive to have uptime rates of “five 9's” or more. In this case, worrying about the SLE may not be an issue, since the ARO is already required to be at a set level. But it does not mean you should place all your eggs into one basket, to speak of. Sometimes the SLE of an event could become so high, if it does happen, critical damage to the company could happen, such as financial data for a bank. In this cause, you must try to find a way to limit the damage caused. It's a trade off in every aspect, so keep in mind the benefits of looking at the problem from every angle. You may find out you've missed a few easy fixes.

This principal, lowering the damage done, or how often damage is done, applies to most aspects of security. Lowering the damage an attacker can do will often pay off better then trying to keep them out. The reason is simple, when an attacker does get in, the damage could be critical to the company. You may have the ARO raised to the point the event will happen once in a billion years, but you can never be sure that the one in a billionth year won't be today. And if it takes all of your assets with it, then tell me just what is the ALE of Infinity x .000000000001 is?

Encryption and Digital Signing

Note: This part is just a quick overview on how Encryption and Digital Signing works. It won't actually show you any useful information for locking down a network, but it will help you understand how these technologies work. Again, this is a very light covering of the topic.

When people talk about security, Encryption and Digital Signing are two issues that are tossed around with wild abandon. Encryption is used to prevent others from seeing information if they gain access to it. Signing is used to detect if a document was tampered with, but it does not actually encrypt the data itself.

Encryption in itself can be one-way or reversible. Reversible encryption means that if you have the encrypted data, it could be decrypted to show you the original data. One way encryption makes what's known as a Hash. A Hash is a set length representation of the data. For example, a CRC Checksum is a 32-bit hash.

To see how these works, here is a very simplistic example. Say you wished to encode the word cat. To use a reversible encryption, you could convert each letter into it's numerical value. In this case, just using their ASCII values, it would be 99, 97 and 116. The encryption would then process these numbers, in this case we'll add 1 to their values, for 100, 98, and 117. This gives us a final encryption of dbu for the word cat. Now if this was to be encoded using a 8 bit hash, it would have a value of 0 to 255. So for our value, we will add these numbers, and “roll” these values back to 0 when they get greater than 255. So 100+98+117 = 315. 315-255 = 60. If you saw this hash in it's ASCII form, it would look like a <, the ASCII character with a value of 60.

What does this mean? Well, most passwords are stored as Hashes. When the user enters his or her password, the system hashes your password, and sees if they match up with the value stored within the hash. Looking at our 8 bit hash example, however, you might see an issue. If the user's password was cat, the person entering in the password would need to only enter an <, or the word das. They would both have the same value in these cases. Two data items that share the same value when hashed are called Collisions.

Collisions happen per hashing algorithm. DES for example, and 3DES, will not generate the same values for words. Also, real hashing algorithms will have much larger hashes than 8 bits. CRC uses 32 bits, but this is too tiny to be of use in security. It just tells you if a file has been damaged, and even then, it's not too precise. A 32 bit hash has 2^32, or 4,294,967,296, possible combinations.

Implementation Topics

Quick overview: What do I need to do to protect my system? ***Home***

Want to know what you need to do to protect your system? Back it up, and keep bad things from getting on it. That pretty much sums up most of the security needed for home users. For companies there's a bit more to worry about, especially when dealing with the servers within the companies. But for home users, the basics are fairly straight forward. And after you take actions to complete these goals, you can add even more security if you wish, but keep in mind it's an area of diminishing returns. And don't think that security is something you must do as soon as you read this. If you went this long without taking all of these steps, then a little while longer won't cause much more harm, if any. Ease into it, and don't do anything rash. Changing the way you do things completely will just raise the chance that you'll be put off by it, and go back to your old insecure ways. And now, here's a quick rundown of the basic steps covered in this section:

1. System backups
2. Firewalls
3. Automatic Updates
4. Browser Security
5. Application Security
6. Non-administrative accounts

The best thing for security is a backup of your system. Most Windows ships with NTBackup, including XP Home Edition (But unlike it's "bigger" brethren, Home Edition doesn't install it by default. It's on the install CD). With a backup of your system, you do not have to worry if you have to replace the whole system, all your work and data is still there. And if you need to reinstall, you only lose the data that changed between your last backup and when the system failed. Since most backup software can be automated, with a little setup backing up a system can be virtually transparent. Simplistic information on backup follows this section, with a more advanced and technical section behind that.

And while a backup is great to have, it's not going to do a lick of good if your system needs restored daily. So you must limit the security holes in your system. The quickest way to plug a lot of security holes is via a firewall. A firewall comes in two forms, a network firewall that sits between all your computers and the Internet, and a client firewall, that runs on a computer and only protects that computer. For home users, a client side firewall is just fine. If you use Windows XP, it ships with a firewall built in, and on SP2 (Service Pack 2), it's on by default. On XP SP0 and SP1, it's off by default. Enabling the firewall is simple, and will be covered in it's own section later, again marked with a ***Home*** marker to make it easier for home users to find. Also keep in mind, most home routers used for cable and broadband Internet are often listed as network firewalls, but in truth are not. They do offer some added security, but they can be bypassed in many cases. They simply do not perform the functions of a true firewall.

Besides the firewall built into XP, there are also many free client-side firewalls out there, a popular one being Zone Alarm. These firewalls also allow outbound "security", in that you can control what applications can connect to the Internet. However, do not be fooled. Just because your firewall doesn't allow your software to connect to the Internet doesn't mean it still can't send or receive data. It simply has to ask another application to do it. Application control in a firewall simply works for controlling applications that play nice and do what they say they do, but you don't want them to be connecting to the network. It's not really a security concern, since it's so trivial to bypass. It can help prevent a worm from directly sending data on the network to infect other machines, but this protection should be handled by the other clients on the network. And on top of that, it would only work as a security function in the event that the system is already compromised.

Also remember, a firewall will only protect against "unsolicited attacks" in most cases. If you go to a site and it has an attack that runs through your browser, or if someone on your yahoo messenger uses an attack against this program, a normal firewall will not stop this. Some of the more advanced network firewalls can protect against this, but only after the attack is known, and a filter is made to stop it. For a normal home user, a firewall will not protect you from damage caused by programs. Each and every application that has Internet access is a potential attack vector against your system.

Another simple security boost is to enable Automatic Updates (AU). Automatic Updates use a Windows service called BITS, or Background Intelligence Transfer Service. In simple form, this allows AU to only download updates using "spare" bandwidth. If you're updating the system, and then go to a website, all traffic for the website will have a higher priority than data for AU, and your web surfing will not slow down. If you're not using all of your speed, than it uses what's left of your bandwidth. The actual background is a bit more complex than this, but to the user it's transparent. The only issues with using automatic updates, if you have 2+ machines on a network, BITS won't detect when ones surfing the web, and ones using AU. While it can be setup to, out of the box it won't bother. The other issue to keep in mind, is the actual installing of the updates. Besides the off setting, AU has three settings, a fully automatic mode where it installs the updates. A download only version I use on my home machines that downloads them, and allows me to click on a little yellow

shield icon on my tray when I wish to install. The third setting just tells you there are updates to download, but doesn't update for you. I recommend running full or manual installs. This will not only keep the system itself up to date, it will also update your browser if you use IE, and your firewall if you use Windows firewall.

And if you're a more advanced user who has more than one machine at home, MS has special server software you may wish to look into, such as SUS and WSUS. The server downloads the updates once, and then the client machines download their updates from this server on your home network. It's nice if you have 3+ machines, since you only need to use up the external bandwidth one time. And there's also the option to manually download the actual .exe installers for all updates. While this can be time consuming, it allows advanced users to "slipstream" the updates onto an install CD, so if you need to reinstall Windows, it'll have all the security patches off the bat.

Once you have a system backup in place, a firewall, and Automatic Updates, what's next? Well limiting how most "bad" bits of software get into your system is a start. And one of the major ways programs and "malware" can get into your system is your browser. So we'll cover this first.

Internet Explorer (IE) has two main security models: Opt-in and Opt-out. In opt-in mode, no site is "trusted", and you must add-in what sites you want to be able to run applications. Opt-out mode, all sites are trusted, and you tell the browser what sites not to trust. However, for all intents, IE ships without these modes enabled. The security setting for the non-listed sites (The general Internet) has too high of privilege to be safe. But the reason it doesn't ship with a higher setting, it will break many sites people use every day. The solution to this is simple, you can raise the security in the Internet zone up, and then any sites you trust, you can add to your Trusted Zones in IE if they do not run correctly.

This process is simple, and involves just a few clicks. Inside any Internet Explorer window, in the top bar go to Tools -> Internet Options -> Security tab, and click the picture labeled "Internet". Now raise the slider up to High. That's all you need to do to lock down IE's security. If you ever hear about all these attacks against Internet Explorer, I would say 90% or more require scripting to run, and this is a conservative example. I personally can think of no attacks that will run without scripting at all.

Now if you go to some sites, you may notice issues caused by the higher security, like blank pages or buttons not working. If it's a site you trust, you can add it into trusted zones. But first we need to set up the Trusted Sites. Again, inside any IE window go to Tools -> Internet Options -> Security, and this time, click the Trusted Sites icon. On the slider, raise the security to medium. Then click on sites, and inside there, you will see a check box marked "Require server verification (https:) for all sites in this zone", and uncheck it if there's a check mark in the box (It's checked by default). Now when you wish to add a site into your trusted zones, you can add it in using this format: *.<domain name>. For example, to add in microsoft.com you would add in *.microsoft.com. The * is a wildcard, meaning it can match anything. So by using *.microsoft.com, www.microsoft.com and us.microsoft.com for example would both work. You just need to add in the domain name, not the address to the page. For example *.microsoft.com/this/page/doesnt/exist.html isn't required, just the *.microsoft.com.

The last issue is running as a non-administrator. When you first log into Windows, it generates a security token, and is passed to the first application ran at login. When you start a new program, it is launched from a program that was already started; for example, using explorer to start up Internet Explorer or MS Word. When these new applications are launched, they get a copy of the parent's token. These programs then have the rights granted to said token. What does this mean? When you run a program as an administrator, the program itself is an administrator. It has all the rights and damage potential as an administrator. And any virus, worm, or other form of attack that succeeds against the program will have the same rights as said program. This is why not running as an administrator is a security requirement. Properly running a system as a non-administrator can be tricky in some cases, however. Later on I will cover some of the details and pitfalls to running as a non-administrative user on your own machine.

Backing up of data: types of backups and data formats. ***In depth***

In security, perhaps the biggest thing you can do to prevent and\or limit the damage done to information is to set up a backup strategy. This not only limits the damage done by loss of Availability, because you can recover quicker, and also allows you to reverse changes in Integrity, by going back to a backup from before the compromise.

There are three major types of backups, Full Backup, Incremental Backup, and Differential Backup. The major difference between these three is what they do with a file depending on it's Archive Attribute. Whenever a file is changed, the Archive Attribute is turned on. This is used by your backup software to determine if a file should be backed up or not, depending on the type of backup you are running. A good backup strategy would be a full back up every weekend, with an incremental or differential backup every day. This means no matter what happens, you only lose one days worth of data, plus any data you would have gained during the recovery process. The differences are:

Full Backup - When you perform a Full Backup, it backs up every file, no matter what the Archive Attribute is set to. It then clears the

Archive Attribute. To perform Incremental and Differential Backups, you need a Full Backup to start from.

Incremental Backup - When you perform an Incremental Backup, it only backs up files with the Archive Attribute set on. It then clears the Archive Attribute, so during the next Incremental Backup, the files will not be backed up unless they have changed.

Differential Backup - When you perform a Differential Backup, it works like an Incremental Backup in that it backs up only files with the Archive Attribute set. However, it does not reset the Archive Attribute.

Why use one type of backup over the others? Well, they all have advantages and disadvantages, and the disadvantages can be limited by using more then one type. You always need a full backup. A Full Backup is slow to perform, and takes a lot of space. But if you wanted to recover from a system issue, the Full Backup is the fastest. An Incremental Backup is the fastest to backup to perform, but may be the slowest to restore from. A Differential Backup is the inverse of the Incremental Backup, its up more space then Incremental Backup, but recovers faster. Remember, Incremental and Differential Backups contain all changes since the attribute bit was changed, via Full Backup or Incremental Backup. Using Incremental Backups daily mean each backup only contains that days backup. Using Differential Backups every day means

The concept of how these three backups come into play is best described via example. Say every Sunday, you perform an automated Full Backup. Every Monday, Tuesday, Thursday, and Friday, you perform a Differential Backup, and perform no backup on Saturday, due to the offices being closed. And every Wednesday, you perform an Incremental Backup. Each day, 50 megabytes of data is changed, and the total data on the server is 1 gigabyte. Well, every Sunday, you make a backup of 1 gig in size, with every bit of info saved, and then reset the Archive Attribute. On Monday, you perform your Differential Backup, and back up the 50 megs work of changes, but do not change the Archive Attribute. On Tuesday, you perform your second Differential Backup. This backup will record all changes to the files since Sunday, so in this case, it will back up 100 megabytes, not 50. Wednesday comes along, and you perform your Incremental Backup. This requires you to back up 150 megabytes of data, 50 each from Monday, Tuesday, and Wednesday. But on Thursday, you only back up the 50 megabytes of files that changed since Wednesday, and your Friday backup is 100 megabytes.

Using this rotation, you have the following backup sizes:

Day:	Backup Type:	Backup Size:	Data Stored:
Sunday	Full Backup	1024 Megabytes	All Data
Monday	Differential Backup	50 Megabytes	Monday
Tuesday	Differential Backup	100 Megabytes	Monday\Tuesday
Wednesday	Incremental Backup	150 Megabytes	Monday\Tuesday\Wednesday
Thursday	Differential Backup	50 Megabytes	Thursday
Friday	Differential Backup	100 Megabytes	Thursday\Friday
Total:		1,474 Megabytes	

Now then, say you need to restore the server to Tuesday's backup. First you would apply the last Full Backup, Sundays. Next, you would apply Tuesday's backup. If you wanted to restore to Friday, you would apply your full, then Wednesdays, then Fridays. Another rotation would be all Incremental Backups during the week:

Day:	Backup Type:	Backup Size:	Data Stored:
Sunday	Full Backup	1024 Megabytes	All Data
Monday	Incremental Backup	50 Megabytes	Monday
Tuesday	Incremental Backup	50 Megabytes	Tuesday
Wednesday	Incremental Backup	50 Megabytes	Wednesday
Thursday	Incremental Backup	50 Megabytes	Thursday
Friday	Incremental Backup	50 Megabytes	Friday
Total:		1,274 Megabytes	

This backup strategy has the smallest total size, and the backup is performed the fastest during the week, only needing 50 megabytes per backup. The disadvantage in it is recovery, while recovering to Monday is just as fast as recovering to Monday using a Differential Backup, it gets slower by Friday, since you not only have to recover from Sunday's Full Backup, but also restore every single day up to Friday.

And the last format, all Differential Backups during the week:

Day:	Backup Type:	Backup Size:	Data Stored:
Sunday	Full Backup	1024 Megabytes	All Data
Monday	Differential Backup	50 Megabytes	Monday
Tuesday	Differential Backup	100 Megabytes	Monday\Tuesday
Wednesday	Differential Backup	150 Megabytes	Monday\Tuesday\Wednesday
Thursday	Differential Backup	200 Megabytes	M\T\W\Thursday
Friday	Differential Backup	250 Megabytes	M\T\W\Th\Friday
Total:		1,774 Megabytes	

This type of backup uses up the most space, but it is the fastest type to recover from, baring Full Backups daily. If you need to recover Fridays data, just pop in Sundays tape, then pop in Fridays tape.

What format should you use? That's up to you. How often do you need to recover? And how quickly do you need to recover? Also, keep in mind that Differential Backups assume that each day, it's 50 megabytes worth of different files being changed. Chances are, the same file was changed more then once. Because of this, the final backup on Friday will be smaller than you would have expected.

Now that you know about the types of data backups, there's the media to worry about, and what to do with the backups. In general, this will be completely different from a home user and office worker backing up a server.

If you're just a home user, then copying the backups to a different drive may be all the security you wish to have. While this won't allow you to have protection against a virus, actual virus infections are rare, most attacks are simple worms that do not infect files. And of the cases where you do get infected, most viruses only attack executable files, so a compressed backup would be safe. This won't protect you if a power surge takes out all HDDs in a machine, so it's up to you how critical the data is. Also, most data like MP3s and pictures\video files, they NEVER change. If you back them up separate from your normal backups, you can make just one Full Backup, then a few Incremental Backups as you add more files to the folders. If the primary concern for a home user was HDD failure, then a RAID-5 system might be the best. By using three or more drives of the same size, you lose one drive as a "backup", but can withstand the loss of any one drive. It also gives you improved read speeds, working similar to a RAID-0 drive. Write speed depends on a lot of variables. If using software based RAID-5 on a slower machine, write speeds suffer, but on many hardware based solutions, write speed is the same as a single drive, and can even improve.

If you're planning a backup strategy for a business environment, the requirements change, due to the added resources at your disposal, and less value placed on data. Depending on where the user profiles are stored, and what a user may store in their user profile, you may not even need to back up local machines. Most companies do not have a need for desktop machines to be backed up. If the machines are just data terminals, then a reinstall with no restore is often the best bet. Servers with data on them, however, must be backed up regularly. And the backups will generally be kept for a set length of time, depending on the server. Your primary SQL server, for example, you may never overwrite a backup tape from it, but rather keep them all as archives to be restored if older data from a set time frame is ever required.

Please note, while this section is rather large, it has just scratched the surface of backups. Topics such as Tape vs. Optical vs. Disk based media, off-site backups, network-based backups, and physical security of the backups (A major issue in businesses, easily one of the biggest issues for many companies) are all things you should look into on your own. Debates on the value of encrypting the backups abound, and are also not covered here.

Slimming down the attack surface: Firewalls

While backups can control the damage done to a system, you need some sort of security to keep the majority of attacks from getting in. This is where firewalls come into play.

When looking at an objects security, you have a limited number of ways in. This is called the Attack Surface. When dealing with automated attacks, a smaller attack surface is better, since it limits the number of attacks that will work, causing many worms from getting into the network. But against a determined attacker who knows what he's doing, these are generally just going to delay him, not stop him., because all it takes is one path in to compromise the system.

Even with that doom and gloom there, a firewall is one of the most important features of security. There is nothing that can shrink down the Attack Surface of a networked machine better then a firewall. But firewalls come in different types, and perform different functions, and can be configured and deployed in so many different ways, it's mind boggling. And many people seem to think of devices that are not true firewalls, such as your NAT-enabled router, are protecting them, when in most cases it's trivial to bypass them, mostly via blind spoofing attacks.

Types of Firewalls: Network vs. Client

Note: This is a very basic, very quick overview of firewall types. It's recommended the reader looks into more in-depth information on the subject. The firewall is the single most important step in controlling your attack surface, and it should be considered a mission-critical item.

Most firewalls fall under two types of classification, Network Firewalls and Client Firewalls. A Network Firewall goes on the outer-edge of a Zone, or Network, and protects it from outside attacks, while maybe also preventing internal traffic from making it outside. A Client Firewall runs on a local machine, and protects the local machine from attack, and possibly protecting other systems from attacks by it. It's important to keep in mind that these two types of firewalls perform different duties, and should be configured as such. But a Client Firewall can be used as a Network Firewall if required. This is generally done inside home networks, where a dedicated Network Firewall would have crossed the line of diminishing returns.

Network Firewalls

When using a Network Firewall, your goal is to protect the network behind it from attack, and maybe to keep sensitive information on the inside from leaking out. A Network Firewall would be any firewall at the entry point to a network (at the default gateway\gateway of last resort). Network Firewalls come in many types of configurations, but there are three main configurations of Network Firewalls talked about in this paper: Single Firewalls, Three-Pronged (also called Three-homed), and Demilitarized Zone Firewalls (DMZ).

The classification of these firewalls is simple. A Single Firewall is just simply one Network Firewall preventing access to the network, and contains two sides: Internal and External. A Three-Pronged Firewall is a firewall with three sides, not the two of a traditional firewall. It contains the Internal Network and External Network connections, but also contains a third connection. This third connection is used for Public Servers, and generally as less stringent security rights on data heading to the servers then data headed to the internal side of the network. It should be noted, many home-based routers use the term DMZ in place of a Three-Pronged Firewall, but they are not true DMZ setups. A DMZ is not actually one firewall, but two network firewalls in a row, one connecting to the External Network, and the actual Demilitarized Zone itself, and a second firewall connecting from the Demilitarized Zone to the Internal Network. The Demilitarized Zone between the two firewalls is where you would place Public Servers. This setup allows you protection for your Internal Network if the first firewall is breached, or is one of the Public Servers are compromised. It should be noted, of these three configurations, there is only one firewall at all times between Public Servers and the External Network.

When configuring your Network Firewalls in a DMZ setup, it's recommended that the two firewalls be of different brands. This is in case one of these brands can be bypassed by a vulnerability, you still have the second firewall up to protect you. And because different Network Firewalls have different abilities, this allows you more control over your network. Examples of Network Firewalls would be Cisco's PIX, Microsoft's ISA, Linux's IPTables, and the offerings from Checkpoint.

Client Firewalls

Client Firewalls, unlike Network Firewalls, are just to protect the local resources on a device. They generally offer more options for outbound control then Network Firewalls, but do not have the capabilities of their "big brothers". A Client Firewall isn't as important for security as a Network Firewall, and they should not be used in place of them, with a few exceptions. But a Client Firewall is a nice piece of software to run inside the network, as it can prevent compromised devices from compromising other devices on the network. This can limit the infection of a Worm or Virus from spreading to other machines, and add in the difficulty of an attacker moving from one machine to another. But because remote administration can be blocked by their use, they add one more layer of complexity that may be of diminishing returns to the user.

In a home network, Client Firewalls take on new importance. Because few people would wish to configure a dedicated Network Firewall, they run a Client Firewall in its place. Sometimes, they have one primary machine connected to the External Network, and it uses it's Client Firewall to block any attacks to itself. All client computers then connect to this machine, turning the entire first device into a Network Firewall. However, since this first computer has more functionality then a traditional dedicated firewall, there are more holes in its Attack Surface. Because setting up a dedicated Network Firewall can be as simple as downloading IPCop and placing it on an old

computer, this practice of using a Client Firewall isn't recommended, but for a home network, it's value on returns is debatable.

It should be noted, most home routers are not valid Network Firewalls. Most of these devices are simply routers that perform NAT. Many of these devices will route packets from the External side to the Internal side, even when not desired. For example, say your internal network is 192.168.0.0 network, and the External interface of your router received a packet with the source IP spoofed to be 192.168.0.8. Many home routers would route this packet into the internal side. To make things even worse, many Client Firewall would trust all clients on the local network, allowing this packet to bypass almost all home security. Because IP Spoofing is a Blind Attack, the value of this attack is somewhat limited, but the threat does exist. And because many ISPs have large Local Loops, there is a surprisingly large number of attackers who could perform such an attack. Remember, in many cases, you're only as secure as the person next to you.

Another role of Client Firewalls is application control. Most Client Firewalls block inbound and outbound connections on a per-program biases. And while inbound connection is great and needed, it's the outbound protection people seem to think is the major function these firewalls should be providing. And while outbound control of applications is nice to have, unfortunately it is not the security blessing one would think. The reason is simple: if a malicious program wishes to send data outbound, it just has to use some other, trusted process, to do it. And there are so many processes to use on Windows for this, it's child's play. Even a simple call to Internet Explorer, with the information encoded in the URL request, is all that's needed. The worm can perform this call without showing the Internet Explorer window, so that the user of the compromised machine will never even see it.

And if you were wanting a Client Firewall to prevent worms from spreading, then you would have to already be compromised. If the system is infected by a worm, chances are it has administrative rights by now (else most of the time it would have failed to infect the machine in the first place). If a worm has administrative rights, it has the same rights as you. Including control of anti-virus and firewall programs. The only time outbound control works on a Client Firewall is when the application plays nice and does what it should (Or is a poorly crafted piece of malware). So while outbound program control is nice, it isn't really a security concern, since the very ideas behind it can be bypassed with ease.

Security in the Microsoft Windows operating system.

Note: Most of this applies only to the Windows NT line, such as NT, 2K, XP, and 2K3. Some may even only apply to XP and 2K3. Anything just on 2K3 and XP will be marked.

How do you limit security issues on a Windows system? Well, first the network it is on must be secured. Sometimes however, this can't be done. If you're a home user with a cable modem and router, chances are you're not going to buy two hardware firewalls for a DMZ configuration. So you have to make due.

To secure your Windows systems, we'll start with some of the biggest issues, and work from there. Firewalls have been already covered at the network level, and backups have also have been covered, so these are assumed to be in place. Some information directly pertaining to Windows' built in firewall will be talked about later, but for now we'll cover some new topics.

Running Microsoft Windows under a Least User Privilege account.

Note: Microsoft Vista has changed how user accounts work when using administrative rights. For example, Vista will prompt you for an administrators name and password automatically when a program performs an action requiring administrative rights, making many of these lock-down tasks simpler. It is still recommended to not run Vista as an administrator, but the potential damage done for it is lessened.

One of the most important steps in running a secure Windows device is to not run programs while you're on an Administrative account. In Windows, when you log in, it creates a security token. This token has your user name, lists what rights you have, and what groups you belong to. After the token is created, any application spawned from the application holding the token will use a copy of the original applications token. Any application my drop rights from it's token, or generate a new, lesser token to pass on to other applications, to limit what they can and can not do. But you generally can not add rights to a token, you need to authenticate to the system, and be handed a new token. This is why when you add yourself to a group, you need to log off, and back onto the system, to generate a new token. In truth, this is only partly true, you only need to log in, you can actually skip the log off part. This trick is used later in the MakeMeAdmin.bat file that will allow you to generate a command prompt running as your current user, but as an administrator. And do not worry, this batch file is not a security risk, you still need passwords to do this.

For home users, setting up a least-user privileged account, in other words, a normal, not administrative-user account, seems like a daunting task. And it's generally thought of that most programs won't run as non-administrators. I actually can think of no programs that will not run as a non-administrator, baring programs designed for administrative functions. Installing said programs is a different story, mind you, but that's easy to overcome. Nero Burning ROM for example can not be used by normal users, but they have a workaround program, Nero Burning Rights, that allows you to assign permissions to users to burn the CDRs. This is not recommended for businesses.

The simplest way to change to being a least privileged user is to make a new administrative account, if you do not already have a new one. This new account must have a password, and not just because of security reasons, but due to built-in security in XP and 2K3 that will prevent you from using it in our examples. Accounts with no passwords can not log in remotely, and using the run-as command uses terminal services, and acts like a remote log-in. Once you have made this new account, you simply remove your account from the administrative users group (Or set it to a limited account, depending on the terminology you wish to use). Now when you log on, all applications should still work the same. One major program that may give you trouble is Nero Burning Rom. It actually has a work-around though, an application called Nero Burning Rights. Download it, run as an administrator, and it will allow you to grant rights to your normal account.

When running as a non-administrator, you now have two problems, installing programs, and performing system tasks. And, a few rare, older programs toss in a new problem: Settings that are set per-user, but can only be changed by an administrator. While you could always log in as an administrator to perform installs and system tasks, a simpler way would be to give yourself a way to make your own user account an administrator. This is where the batch file MakeMeAdmin.bat comes into play. This batch file is as follows:

```
setlocal
set _Admin_=%COMPUTERNAME%\<ADMINISTRATIVE ACCOUNT>
set _Group_=Administrators
set _Prog_="cmd.exe /k Title *** %1 as Admin *** && cd c:\ && color 4F"
set _User_=%USERDOMAIN%\%USERNAME%
if "%1"=="" (
        runas /u:%_Admin_% "%~s0 %_User_%"
        if ERRORLEVEL 1 echo. && pause
) else (
        echo Adding user %1 to group %_Group_%...
        net localgroup %_Group_% %1 /ADD
        if ERRORLEVEL 1 echo. && pause
        echo.
        echo Starting program in new logon session...
        runas /u:%1 %_Prog_%
        if ERRORLEVEL 1 echo. && pause
        echo.
        echo Removing user %1 from group %_Group_%...
        net localgroup %_Group_% %1 /DELETE
        if ERRORLEVEL 1 echo. && pause
)
endlocal
```

To use this on your own system, you must change <ADMINISTRATIVE ACCOUNT> (Second line) to your administrative account's name. When you need to be an administrator, rather then using runas for the new account, you run this batch file. It can be a bit of a pain, you need to enter in the administrative account's password, then your own password (both accounts must have passwords for this to work). After completing this, a new CMD window pops up. It is running as your user account, but it is in the administers group, and has administrative rights because of it. Now the thing to keep in mind, when one program (Such as CMD, or Explorer) launches a program, it generally copies it's own security token to the new program. Any program launched from this CMD window is ran as an administrative user. If you need to install a program, or make changes as an administrator, run this batch, then launch the program. If you need to click on something as an administrator because there's no command line application that you know of to do it, such as something in the control panel, it's easy. Run %programfiles%\Internet Explorer\IExplore.exe and type the address of the program. Want to set permissions without using cacls? Enter the address into IExplore's address bar, such as c:\program files. To get to the control panel, just type Control Panel in the address bar. From there you can run any applets.

Too much work on your low physical security home machine? Get an application that runs in the tray, and has a file command. One is taskmgr, built into windows, so we'll cover this one. Launch taskmgr from the administrative CMD window, then in view, select hide when minimized. Then you can hit file, new task any time you want an administrative window, no need to retype passwords. This is not recommended on a production machine, but for a home machine it works just fine.

Locking down the browser: Internet Explorer Configuration

Internet Explorer is the default browser for Windows. People talk a lot about security issues with Internet Explorer, but they generally do not have a full understanding of the concepts. Then generally quote the number of bugs. If you've ever looked at CERT, one thing you should notice off the bat: many IE security issues come out after, not before, there's a patch. Indeed, there's many that can only work on XP SP1 or before showing up a year or two after SP2 was release. And this doesn't take into account almost every security issue with Internet Explorer can be mitigated with ease. What good is a security breech that can't read or write anything?

The first thing to do, you should have already been doing from the previous sections. Stop running Internet Explorer as an administrator. When you're not an administrator, you do now have NTFS write permission to c:\program files or c:\windows (Or whatever these folders are called on your system). Internet Explorer, even if compromised, should only be able to write to your user profile. This right here stops many forms of spyware, with no interaction on the part of the user. But it wouldn't be any fun if that was all we could do...

If you have ever tried any exploits for Internet Explorer, you'll know that most of them require scripting. Even the .WMF image issue in 2006 needed scripting to get it to work when ran inside Internet Explorer. It mostly only worked right when the image was stored on the local machine. I'd say a good 95% of attacks on Internet Explorer require scripting. While they may not be script based, they still need it to run. While this number is just an estimate off the top of my head, of the 5% of issues that do not need scripting, it's doubtful you will see them take place before you can apply the patches, and even then, would they work when not an administrator? And what could they even do from such a limited breach?

To disable scripting in Internet Explorer, just go into tools -> Internet Options -> Security, then click the Internet option. Raise the slider up to High. Hit the ok button. That's all there is to it. If you want, you can then hit custom level, and change a few fine-tuned security levels, such as allowing downloads, so you can still save files from sites.

To make Internet Explorer more usable, while on the Security page, set Trusted to mid. Then press the sites button, and uncheck the Require Https box. Now any site that does not work with the new security, you can add to trusted sites by entering them here. In 2003, IE7, and a few other versions, the current site will show up in the sites box for you, otherwise you have to manually type in the site.

While on the topic of Internet Explorer, we should cover what Internet Explorer really is. Internet Explorer is not, like most people think, a program. When you go to a site, it's Explorer, plus a few Internet API calls. When you update Internet Explorer, you're actually updating this API, an API used my more applications than you think. The Internet Explorer API is used by items such as the HTA subsystems, the compiled help system, most control panel extensions, the MMC console and all it's snap-ins, and many others. Even third-party applications use it. Heck, even your desktop uses it if you have anything other then a .bmp image. The jpg support is part of the Internet Explorer API. The settings in Internet Security apply to anything using these API calls (Such as Windows Media Player, and even some codecs) So even if you do not use Internet Explorer as your primary browser, it would pay off to set these security settings, since they apply to many other programs.

It should also be noted that all these Internet Explorer settings can be controlled via local and group policy.

Rights and permissions

In this section I will be covering how rights and permissions work on a Windows machine. Permissions are set on an object, and enforced by an ACL list. For example, you may have the read permission on a file. Rights are set on users or groups. These are generally global items, such as the right to shut down the system, or to take ownership of files. Most of the time when you set rights, it's done via a GPO at the domain level, and not directly onto the user account. In my personal experience the two most common rights that are modified would be the rights that grant the power to shut down a PC, and that allow you to log in locally onto a machine (Such as allowing a normal user account to log into a domain controller).

Allow, Deny, and the other one...

Many people who work in security have a Unix background, or use a home machine that's Linux based, and have a hard time grasping how permissions work in Windows. In the basic form of Unix file permissions, the permissions are Read, Write and Execute, and can be set to one of two levels, allow or deny. On a Windows machine, there are many many more permissions, and you have three levels to each permission, Allow, Deny, and "Revoke\None". While it seems like Deny and Revoke may be redundant, they do operate in completely different manners. Revoke simply means there is no permission granted, but it also say the user doesn't have access to it. If the permission set on the item is "Revoked", it checks the objects parent to see if it has permission. It continues flowing up this tree until it finds the root object, and if there is still no allow or deny, then permission is not granted, or in other words, denied. The best way to show how these work would be with an example as follows.

Say you have a file called Test.txt, located within c:\TextFiles. If a user tries to access Test, the computer will first check to see what the user's access is on the file itself. In this example, if the administrator set an allow or deny directly on the file, then that would be the users permission. But in most cases there will be no explicit file permission set on files directly, and so the system will then check the next object up, the folder TextFiles. If the user has been granted permission to the object here, or denied permission, then the system will allow or deny access respectively. If at this level there is still no permissions granted, then the system checks the next level up, the partition itself, in this case C:. If the user has permissions here, the system grants access to test.txt. If the user is denied, then no access is granted. If the user still has no direct permission, then the system denies access. The root of a drive is the highest object in file permissions, and there is no "computer" security, so once it hits the root of the drive, permissions are decided one way or the other.

When working with permissions, it's possible to have both an allow and deny at the same level. You may be a member of the Human Resources group, who have access to some sensitive records. But you may also be a member of the Temp Employee group, who have a deny on the same records. In cases like this, the deny will override the allow, and access is denied. It doesn't matter if you have 10,000 allows, 1 deny from the same level will block it.

What happens if a user is denied access to c:\NoRead but is allowed access to c:\NoRead\YesRead? Well the user can access YesRead just fine. The deny on the higher object doesn't come into play because the allow is closer to the object.

One more item of importance to mention, an object can be set to not inherent permissions. When you chose to do this, you will be prompted to copy all current permissions, or erase all current permissions. From then on, when you set permissions on this object's parent, it will no longer propagate down to it. For example, if c:\Test\Folder was set to not receive permissions from it's parent, any changes to c:\Test's permissions would not carry over. This can cause many nightmares if abused, so be careful using it. Since you can assign a deny or allow directly on the object, there are few cases when you will need to disable permission propagation.

Besides file systems, permissions work like this for shares, printers, SQL objects, registry items, and even group policy objects. Understanding permissions will not only give you control of what files one person can access, but you can use group policy to actually only work on groups, not just OU's.

Making your own programs support Windows security models.

There is a lot to learn about Windows security if you wish to administer a system. If you want to make applications for Windows, you should understand Windows security, in addition to being able to program. But it is a rare person indeed who programs for a living, and understands system security. Hence, so many Windows applications break simple security rules, because they think the way they're doing it is better, without knowing about the items they break in doing so.

This section will cover one tiny part of application security in Windows. It's not talking about how to make your application crash proof, or how to prevent buffer overflows, or how to protect your heap. This is about where you should save your data, and why.

Many applications pride themselves in not modifying the registry in Windows, instead, they save all data and settings inside the programs subdirectory. While this is nice for smaller applications, it actually causes complications for administrators, and has security related issues that can prevent the application from being deployed over a network, and adds extra work for the administrator.

You may be wondering, why would having the program store info in less places create more work? Simple, permissions, and per-user settings. When a program is installed in the Program Files folder, normal users only have read and execute permissions within this directory. This is designed for a few reasons: prevent users from changing applications, prevent users from installing applications, and prevent viruses infecting a user process from changing the contents of the Program Files folder.

To understand how Program Folders is intended to work, you need to keep in mind, data and settings for programs should be per-user. Global settings should only be changed by administrators. Because normal users have no right to write to the Program Files folder, no program should ever write to this folder once installed. The only exception would be to store global settings, but these should be stored within the registry, for reasons to be covered soon. If the application must store data that all users can access, you will need to create a new folder just for it, directly modify the security on the file it saves it to, or run the process as a service or another user. All of these options create headaches for administration and open up new security issues.

All data from an application should be stored inside the users profile by default. If you want data from an application to be shared, the user could place it inside their shared documents folder, if they have one. Otherwise, a share or folder would need setup with permissions to allow the correct users access to it. Learn how to find the users profile (It's not always c:\documents and settings\<username>) and use this for the default location. And make sure your program can handle the error if the user tries to save, and the save fails.

Now you know where data should be stored, what about settings? Settings should always be stored inside the registry. Why? Because it is much easier to apply changes to settings in the registry then via files. Besides being able to apply .reg files, using the REG command line, and hand editing local and remote machines, the Windows Registry has one very powerful editing mechanism: Group Policy. Ever edit anything within Local Security Policy snap-in? Most of these options are stored in the Registry. The local security policy snap-in just makes some of the most common items easier to see. But that is just a GUI, it doesn't actually add any sort of power. The real power comes when you save these changes into a security template, and\or place them within a Group Policy Object (GPO). Security Templates are registry edits, registry permissions, and NTFS permissions. You can then apply them to the local machine, giving you a stored baseline. You can also audit machines against your security templates, to make sure the settings you say should be in place, are in fact the settings in place. While this is a nice function, it pales in comparison with the power of Active Directory and Group Policy.

A Microsoft Windows Domain is known as an Active Directory (AD) Domain. Active Directory is a powerful tool, and has been one of the major reasons Microsoft has such a strong group on the desktop market for businesses. Active Directory has within it containers, called Organizational Units, or OUs. OUs store User Accounts or Computer Accounts, and can be used to group together people and machines that need settings the same. You apply group policy to OUs, and it is applied to every user and device within the OU. This means that anything you set via Local Security Policy can be applied to one, ten, or even ten thousand machines, with the same amount of work. But it gets better.

The options in local security policy, and in Group Policy, are simple text files that you can make on your own. This means you can add in your own Group Policy, and allow your program to be maintained via Group Policy.

To learn more about the Microsoft Windows Registry, and how you can make your program Group Policy compatible, please see the document http://www.security-forums.com/viewtopic.php?t=34277 Also note that with the coming of Vista, the format for .adm files have changed. And while it does offer many advantages (Mostly file size) it also makes them very wordy, since they are now based on the XML markup language. The MSDN website will offer more information on this improved format.

Diminished Returns: Giving your security the final twist.

If you want to lock down Windows some more, some of the more commonly mentioned security procedures will work, but the person implementing them needs to understand fully what they do. And to make matters worse, not only do some of these procedures require a substantial amount of work, many times you may make things worse by implementing what you think is a better security policy than what's shipped out of the box.

Services, to disable or not to disable, that is a corny section title.

One of the most common bits of security advice for Windows is to turn off unneeded services, but I do not recommend doing this for most people. Now don't get me wrong, I am not in any way, shape, or form saying unneeded services are not a security issue, cause they are. What I'm saying is, few people understand what they're doing inside services.msc and may cause more security issues than they had before.

Services can be set to three modes, automatic, manual, and disabled. Automatic means they start when the machine is started, manual means they start when the user, OR AN APPLICATION, tell them to start, and disabled means they're prevented from starting. Manual services can not start by requests from remote machines (Unless said request is to a program other than the service, and that programs starts the service, but this is rare example). A manual service will only run when needed, and in most cases, it will turn off on it's own. This means that a service set to manual is only vulnerable to attack if an attacker with local access to the machine causes it to start and then attacks it, or an application needs it, and attacks it from there.

By now you should see the main point though, a manual service only runs if needed. Sometimes a program will try to start it to use features of the manual service, but if it can't start it, it'll go about it's work in a different way, or with some functionality disabled. But a service set to manual is not that much of a security risk. And it gets even less as you read more about them.

Even if a service is as risk, an attacker shouldn't be able to get to it in most cases. It should be firewalled, via a client and\or network firewall, from it's potential attackers. So even if the service is active, they should only be vulnerable to "trusted" machines. The messenger service is a prime example. Many people say disable the messenger service, I say, enable it. If you ever do get spam on the messenger service, then it means your firewall is not blocking connections to this port, and it also means it most likely isn't protecting any other services.

Disabling services can limit the attack surface, but this attack surface should have a firewall over it anyways. Disabling services also does not free up as many resources like most people think. Most unneeded services would be set to manual, and thus not start. And if the services are started, but not being used, then their ram will be swapped out into the swap file if another application needed the ram. Because Windows only needs to access the swap file when the ram is accessed, programs sitting idle have almost no performance penalty when their ram is in the swap file.

When worrying about services and security, the item you should worry about isn't if a service is or isn't running, it's who it's running as. On XP and 2K3, services run under many different accounts, from Local Service account to System, there's a bunch of them. Why? Because these services run under accounts that have different rights and access controls. In Windows 2000, all services ran under one account, with almost no security restrictions. And even on XP and 2K3, third party services may install themselves with too much power for what they need to do. This is where you should aim your attention to.

In the MMC snap-in for services (Found in the administrative tools), you will find a list of all your registered services. When you check the properties of a service, you get the start up configuration page first thing, the most commonly seen tab. The log on tab may well be the least seen tab of the group. This tab controls what account the service logs in as. By using both NTFS permissions, and granting rights, you can control EXACTLY what each service can do on your machine. On an XP or 2K3 machine, it is not recommended that you mess with the log-on on default services. They are, for the most part, locked down pretty tightly. It's only 2K and 3rd party services that need locked down. And finding out what access each of these services need can be a true pain.

In summery, do I recommend you mess with services? Not on XP or 2K3 machines. On a Windows 2000 machine, you should look into it, but only make changes once you fully understand what a service does. If you disable a service, and later add an application that requires said service, few applications will tell you why they are failing. They assume that since x service is on the programmer's machine, x service is on all machines. In all likelihood, the programmer didn't even know that what he was doing required a service to begin with. I couldn't begin to count the number of computer issues I've fixed by enabling a service someone disabled. Just a quick glance will show you thousands of "service faqs" that say stuff like "There's no reason for the Server Service to be running on a home machine". Then a few weeks later the user wonders why they can not make shares on their machine. I'm not saying don't mess with services. I'm saying, read up from Microsoft directly, and make sure you fully understand what a service does before disabling it, and please, only disable it on a test machine for a while, not a production server. Microsoft also has lists of service settings for many server configurations. I highly recommend reading up on these, you may be surprised at some of the circumstances some lesser-known services are required.

Appendix

Common Security Misconceptions: Unlearning what you know.

There are many misconceptions in information technology security. Some ideas are just plain wrong, some were good but have since become outdated, and others are just good ideas with poor logic behind the implementation. This section will be covering a few of these misconceptions, with hopes of clearing some of them up.

Misconception #1: Biometrics are the best security development ever - Biometrics are the worst security idea ever.

In security, you will find two opposing views on the topic of Biometrics, those that love it, and those who hate it. The ones who love it are generally the ones who are new to security, and have learned most of it from mass media. They think it should be used to replace passwords and pin numbers as a way for granting access. The ones who hate it are the ones with slightly more info, and have learned it from peers with little to know security background. The most common complaint is the lack of ability to change "passwords". So, the truth of the matter?

The truth is, Biometrics are a great asset to security, and will used more and more over the upcoming years. It's just that the implementation of biometrics is different than what people think. Biometrics is not a password, and it should not be used as one. Biometrics is used to prove that the person entering the password is the one given the password.

Most authentication systems use Usernames\Passwords, or smartcards, or some other method that works on the same principals. The problem with username and password combinations, they are stored just as knowledge, and can be freely given away. While you can prove that it was X person's account that was used, you can not always prove it was X person who did it. Smartcards are slightly different, they are physical items. If you give one away, you lose your own access to the system, and so you are likely to report a missing card rather quickly. Many smart cards have photos of the user on them, so they can be matched to the person using them, but these photos are almost never used by the authentication device itself.

Biometrics allows you to, with a reasonable amount of error, say that X person was the one who accessed the services. You can not give away a finger print or facial scan, and so it helps to proves you are the person you say you are. It is not, by itself, the means of authentication , that should still be performed by a password or pin number.

The biggest problem with biometrics currently isn't the design, as most design issues, such as reply attacks, have solutions. It's the implementation of the biometrics themselves. Some biometrics use anti-reply schemes inside them, where if a scan of the data PERFECTLY matches another scan, it is rejected. This however, is a rare function for most scanners. Another feature is the ability to rescan the stored key, your finger print for example, using different points of reference, creating a new hash value in case the old one is compromised. This only comes into play if the scanner was attacked itself, and the data was sent directly to the authentication device.

Some issues that people quote about biometrics that are not issues is the ability to change the "password". If the "password" is compromised, they think you can not change it. This isn't true, you just have a limited number of changes before you have to reuse passwords: 9 changes to be exact. There is no reason why you must use the same finger all the time, and few networks ever enable a password history enforcement of more then 9 passwords (But the default is 24 in a Windows Active Directory network, so those who do enforce password history will generally have the history higher then 9 before allowing repeats) Also note that the biometrics is not the Secret in the authentication, it takes the place of the user name. User names are not secret, and do not need changed. In many companies, they are generated based on the users name, and are used for the users email address.

Many lesser quality scanners can be fooled via simple tricks, even to the point of photocopied finger prints. It should be noted that tricks such as these only work on cheaper models, and as the technology advances, detection of said attacks will improve. Newer scanners can even examine the pattern of blood vessels using infrared, making simple lifting of a finger print useless.

Misconception #2: Viruses work on one system only. A Windows Virus can not affect a Linux machine.

On March 27, 2001, a virus named W32.Peelf.2132 was discovered. It was a proof of concept virus, not one meant to bring down systems. This virus then also received the name Linux.Peelf.2132, so you might get a feeling where this is going.

While the Peelf was not a threat, and had no native way of spreading, it still could infect files of both the Windows and Linux operating systems (IF they used the Elf binary format, like most do). It could only infect files on the local machine, but it could spread on a Linux machine, or Windows machine, infecting both types of binary files at once.

It wasn't till Win32/Simile that things got messy. The first release of Simile, Simile.A, was a polymorphic virus, one of the most complex to date. This on its own was enough to make it's mark in the history books for viruses. But it wasn't till the fourth version, Simile.D, that things got messy. The D version contained a second infection engine, aimed at elf binary. It would still use the original polymorphic engine when attacking the elf binary, making this attack very hard to detect. Also note this was still just a Virus and not a Worm, and had no way of transmitting itself. But it wouldn't be hard for an attacker to modify the code, and to place it inside the body of a pre-made Worm, causing the Virus to be the actual payload.

Example Configurations

When running a network, you need to clearly define your security goals. One goal you need to define quickly is just how far you should lock down client desktops. My personal beliefs is to lock them down completely, with company wide defined wall papers and the works. Besides easy of auditing, protection from internal issues due to inappropriate wall-papers and the like, you also lessen the chance that a user runs a program that he or she should not be running. Here are a few configuration examples:

Example #1: No contact with the outside world.

Many times, you do not want any outside applications to sneak into a network, or at least onto the workstations without your permission. Most administrators, however, are at a loss on just how to prevent these applications from making it into the systems. With all the different ways Windows allows you to control what programs a user can and can not run, many administrators are at a loss on where to begin. If you want a complete lock down, the way to do it is a lot less complacted then you would think.

When a user is on the machine, the file system itself is partitioned off into different sections. The user's profile is traditionally stored on c:\documents and settings (or c:\users on Vista by default). Inside this profile the user has full control, read, write, execute.

Another section of the drive is the programs folder, again c:\program files by default. A normal user has read and execute permissions on this folder, but should not have write permission.

Knowing this, you must now think, how do programs you do not wish to be ran get onto the system? One way is downloading of files. This can be easily blocked in Internet Explorer and can be published down via Group Policy, or could be configured at install time. Just make sure you do not allow users to change their own security settings. I've known too many companies where end users could modify their Internet explorer settings.

The first part of the configuration needs done on the physical machine itself, and should be done during deployment. As with all machines, the computer should have a BIOS password set, blocking the user from changing settings. The primary setting we want locked out here is the boot order. This is needed to prevent a user from booting to a CD-Rom. The second change may not be usable in many locations. It's disabling of the USB ports completely. USB thumb drives can be security risks that will drive administrators up a wall. By default, anyone can just plug in a USB drive, and have all programs they want right there

Glossary of Terms

0-9

5 Nines – Used mostly when talking about system uptime, 5 Nines, means the system is up and working 99.999% of the time. This generally means total downtime is less then 5 minuets per year, for a 24 hour, 7 day a week, 365 day per year time span.

A

Access Control Lists (ACLs) – An Access Control List is a listing of who, or what, has access to an object, or is denied access to an object.

Annualized Loss Expectancy (ALE) – This is how much, per year, you are losing from this event. The ALE is generated by the SLE x ARO to generate your ALE. For example, with the SLE of $4,000, and an ARO of 12, your ALE is $48,000. If the ARO was .25, then the ALE would be $1,000.

Annualized Rate Of Occurrence (ARO) – This is how often per year an event will happen. If your server goes down once for 4 hours per month, then you have an ARO of 12. If it went down for 4 hours every 4 years, then your ARO for this even is .25.

Archive Attribute – A bit inside the header of a file, on most file systems. This bit tells the computer if the file has changed since the last time it was backed up, or Archived.

Attack Surface – The area that an attack can come through, or, the visible area of a target the attacker can try to compromise. It is the sum of all the possible ways in an attacker can use to get into the the system.

Availability – The amount of time that a service is up and running correctly.

C

CIA – In security, it's also known as The Security Triad. It stands for Confidentiality, Integrity, and Availability.

Client Firewalls – Client firewalls are firewalls that run on the local machine, and are designed to protect only the local machine. They are generally less robust then Network Firewalls.

Confidential (Information) - This is the data used by a limited number of internal users, and should not be known to the majority of workers. This is the class Human Resources (HR) data and Payroll Information falls under. Read access to this data is limited to a few users, and write access is generally restricted even more. If this becomes public internally, Operations and Internal Trusts are at stake, while if reviled externally, you lose your PTR, along with Operations and Internal Trusts. OS files dealing with security also fall into this area in most cases.

Confidentiality – Insuring that data is only observed by people who should be observing it. While this is a major goal in the IT field, having complete Confidentiality can be impossible to insure at times.

D

Demilitarized Zone (DMZ) Firewalls – In firewalls, the DMZ is the area between two firewalls. The DMZ space is normally where servers that need to be accessed from the outside, such as web servers and VPNs, are located.

Denial of Service (DoS) - An attack that prevents normal usage of a service. It could be caused by using up all of your companies bandwidth from the ISP, or purposely logging into an account repeatedly with the wrong password to lock out the account.

Differential Backup – A type of backup that backs up all files with the Archive Attribute set, and does not reset the Archive Attribute.

E

External Secrets – Information held in trust for another entity. Examples of External Secrets would be customer credit card information, or trade secrets from partner companies.

F

Full Backup – A form of backup that backs up every file, regardless of the Archive Attribute. After backup is performed, the Archive Attribute is reset on all files.

I

Internal Trusts – An Internal Trust is a Trust between a company and it's employees, between two or more employees, or even between family and friends. This can be compromised in many ways, from displaying emails that could be considered offensive to some members of the trust, or reveling benefits given to one member not given to another of the same standing: example, payroll information that is often negated on the part of the employee, and isn't purely based on performance and value.

M

MCP – Standing for Microsoft Certified Professional, this is often thought of as being a certification. Being a MCP merely means you passed one Microsoft exam, it does not even have to be a technical exam, most application exams, such as Excel exams, will grant you the MCP status.

MCSE – Standing for Microsoft Certified Systems Engineer, this is the highest level of Microsoft certification for server and network administrators in Microsoft NT, 2000, and 2003 environments. It consists of 7 exams: An exam on desktop OS's , three exams on Server and network (Called core exams), two electives, and one network design exam. Electives generally stem between SQL exams, Exchange exams, and Security exams, but can cover a wide range of topics. The MCSE is a lifetime certification, meaning it never expires, but each updated OS has a new MCSE exam, creating people who are certified in MCSE 2000, or MCSE NT4.

MCSE:Security – A specialized form of the Microsoft Certified Systems Engineer, devoted to the security aspects of Windows. To achieve MCSE:Security in Windows 2000, the user is required to take eight (8) exams, the extra exam being the users choice of a ISA Firewall exam, or the third party CompTIA Security+ exam. The choices of exams leading up to the normal MCSE level is restricted in this path, the user must take a Security exam (70-214 in the 2000 path) in place of one of his or her electives, and must take the 70-220 exam (Again, for the 2000 path) for the design requirement.

R

Risk Assessment - Risk Assessment is the practice of identifying risks to your business or assets, assessing the potential damage done, and recording how often said risk will occur.

S

Security+ - This is the security exam by the vendor neutral organization known as CompTIA. It can be used as an elective for the MCSE:Security specialization, taking the place of a ISA firewall exam.

Single Loss Expectancy (SLE) – This is the value of damage the average event of this type will happen. For example, if the average downtime of a server would be, say, 4 hours, and this down time causes $1,000 in loss per hour, then your SLE for this event is $4,000.

SOX - The Sarbanes-Oxley Act of 2002, also known as Public Company Accounting Reform and Investor Protection Act. This is a United States federal law, it's primary goal is to force companies to evaluate their Internal Controls for finical reporting, and that a third party (In general, an Auditing Firm consisting of CPAs) confirm that reported practices and implemented practices do indeed match. See: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf and http://www.legalarchiver.org/soa.htm for more detailed information.

V

Virus – In computers, a Virus is a program that can “infect” other programs with copies of itself. On it's own, in the purest sense, a virus can not infect other machines with itself on it's own. It must be copied via the actions of users. While it has the most publicity, a virus is still the rarest form of malware: Worms, Viruses, and Trojans.

VPN - VPN stands for Virtual Private Network. It allows two separate networks to act as one while the data from each network is transmitted over a public network, such as the Internet.

W

Worms – A worm is a program that will transmit itself to other devices on it's own, spreading across a network with no user intervention. Worms are generally single files, meaning that many times computer can only be infected by a worm once. Some single-file worms, however, use randomly generated file names, allowing for more than one concurrent instance of the worm to be running at one time.

References used:

Microsoft Course #2050A
Designing A Secure Microsoft 2000 Network

Open Sources: Voices from the Open Source Revolution: Appendix A - The Tanenbaum-Torvalds Debate
http://www.oreilly.com/catalog/opensources/book/appa.html

IPTables Connection Tracking - FTP
http://www.sns.ias.edu/~jns/wp/2006/01/24/iptables-how-does-it-work/?p=20

Linux-PAM modules etc. page
http://www.kernel.org/pub/linux/libs/pam/modules.html

Backup types
http://www.backup4all.com/backup_types.php

Applying the Principle of Least Privilege to User Accounts on Windows XP
http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/luawinxp.mspx

Using a Least-Privileged User Account
http://www.microsoft.com/technet/security/secnews/articles/lpuseacc.mspx

The Services and Service Accounts Security Planning Guide
h*ttp://www.microsoft.com/technet/security/topics/serversecurity/serviceaccount/default.mspx*

MakeMeAdmin -- temporary admin for your Limited User account
http://blogs.msdn.com/aaron_margosis/archive/2004/07/24/193721.aspx

MakeMeAdmin follow-up
http://blogs.msdn.com/aaron_margosis/archive/2005/03/11/394244.aspx

An Analysis of Simile
http://www.securityfocus.com/infocus/1671