



Abstract

This is an ITSS draft document designed to help technically advanced Windows users run under a more secure (non-admin) context. Please send your much appreciated feedback to itss@umich.edu.

Run As User!

You've heard it a thousand times: "*Don't run as admin*". Yet you continue to tempt fate. You log in with admin credentials and surf the wild wild web through whatever minefield it takes you. You open e-mail and attachments with abandon, confident in the fact that you've never been hacked before. Yet every once in a while, your heart starts to beat a little faster. Perhaps it happens when you land on some web site you didn't expect, or when you double-click on that unsolicited email or launch some video clip that your friend sent you. Your heart accelerates because you know, deep down, it's just a matter of time before you do get hacked. And then, because you're logged in with administrative credentials, you know the price could be big. If you're lucky, only your ego will be bruised. Worse, the integrity of your system will be compromised and personal as well as private University information will belong to someone else. In fact, it's entirely feasible that your system has already been compromised and you're not even aware of it. How do you know that it hasn't?

If you're pushing your luck by logging in with administrative credentials, then read this paper. We'll illuminate the "tips and tricks" necessary to start running as user. You'll feel better running in a less privileged context, and you'll be making a critical contribution to the security posture of your unit and the University.

Audience and Scope

This paper is for technically advanced users that log in to their primary Windows system with local administrative credentials because they periodically need to perform legitimate administrative tasks. The focus is on **domain-joined XP/SP2 systems** for the following reasons:

1. If your XP system is **not** joined to a domain, then running as user is simpler and discussed in Appendix A.
2. If your primary Windows system is a Windows **Server** system, then running as user is simpler and discussed in Appendix B.
3. If you have not yet upgraded your XP system to Service Pack 2, then that should be your highest priority.

The techniques described in this paper are **not** appropriate for non-technical "end-users" that log on with administrative credentials. If your end-users are logging in with local administrative privileges, then you should address the root cause of that *serious* security issue. The root cause of that security problem is most likely the reliance on poorly written applications that the end-user must run in order to do their job.

Why should I run as User?

The theoretical justification for running as user is rooted in a fundamental security best practice known as *least privilege*. Least privilege states that every program or system component must operate with the minimum set of privileges needed to accomplish its task [1]. For example, administrative privileges are required to **install** Microsoft Office but not to **run** Office applications. Thus, following the principal of least privilege, you should be an admin when installing Office, but a normal user (non-admin) when running Office applications.

But let's forget about theoretical underpinnings for the moment. The reason you need to quit running as admin is because the hackers are counting on it! When you are running as admin, any piece of code that you launch can [2]:

- Install kernel-mode rootkits and/or keyloggers (which can be close to impossible to detect)
- Install, start and stop services (e.g. stop the Windows Firewall)
- Disable/uninstall anti-virus software
- Install ActiveX controls, including IE and shell add-ins (common with spyware and adware)
- Copy files into Windows directories
- Edit system-wide registry values
- Access data belonging to other users
- Cause code to run whenever anybody else logs on
- Replace OS and other program files with trojan horses
- Access LSA Secrets which may include domain account information
- Modify other local accounts and passwords
- Modify configuration files (such as the HOSTS file for web redirection)
- Cover its tracks in the event log
- Etc.

None of these real-world malware activities are possible when you are running as a normal (non-admin) user. Furthermore, it is important to realize that when we refer to "code", we're not just talking about executables (.exe's). "Code" can take on many forms in Windows. Here is list of some extensions that can result in the execution of "code" on your system:

- | | | |
|--------|--------|--------|
| • .ADE | • .INS | • .PIF |
| • .ADP | • .ISP | • .REG |
| • .BAS | • .JS | • .SCR |
| • .BAT | • .JSE | • .SCT |
| • .CHM | • .LNK | • .SHS |
| • .CMD | • .MDB | • .URL |
| • .CPL | • .MDE | • .VB |
| • .CRT | • .MSC | • .VBE |
| • .EXE | • .MSI | • .VBS |
| • .HLP | • .MSP | • .WSC |
| • .HTA | • .MST | • .WSF |
| • .INF | • .PCD | • .WSH |

Now combine the two lists above. This should make it clear that when you are running as admin, your attack exposure is simply too great.


Why do I currently run as Admin?

Most likely, the reason you log in with administrative credentials is that periodically you need perform legitimate administrative functions such as:

- Download and install applications
- Upgrade or patch the operating system
- Add hardware and install corresponding device drivers
- Update device drivers
- Manage storage devices (Format a new disk drive, Defrag a drive)
- Modify system environment variables
- Configure local security policy
- Manage security logs
- Create shares
- Manage firewall settings
- Perform system backups
- Change system time
- Change network configuration settings
- Etc.

You may also need to run applications that do not work unless you are logged in as an administrator – not because these applications are legitimate administrative applications, but because they are poorly written applications¹. For example, an application that attempts to store per-user preference settings in a per-machine location (such as `HKLM\Software` or `%Windir%\System32`) is a poorly written application that is unjustifiably causing you to run with elevated privileges.

In the end, you log in as administrator because it is too much of an inconvenience to log out and log back in every time you need administrative access. You may have tried using tools such as “Run As” but found that not everything works the way you need it to. The purpose of this paper is to illuminate some “tips and tricks” that will limit this inconvenience to the point where you are willing to log in as a normal (non-admin) user and conveniently switch to an administrative context when necessary.

¹  To identify well-written applications that run successfully under the secure user context, look for applications that have achieved the *Designed for Windows XP* Logo. You can find a list at: <http://testedproducts.windowsmarketplace.com>

Tools for the Job

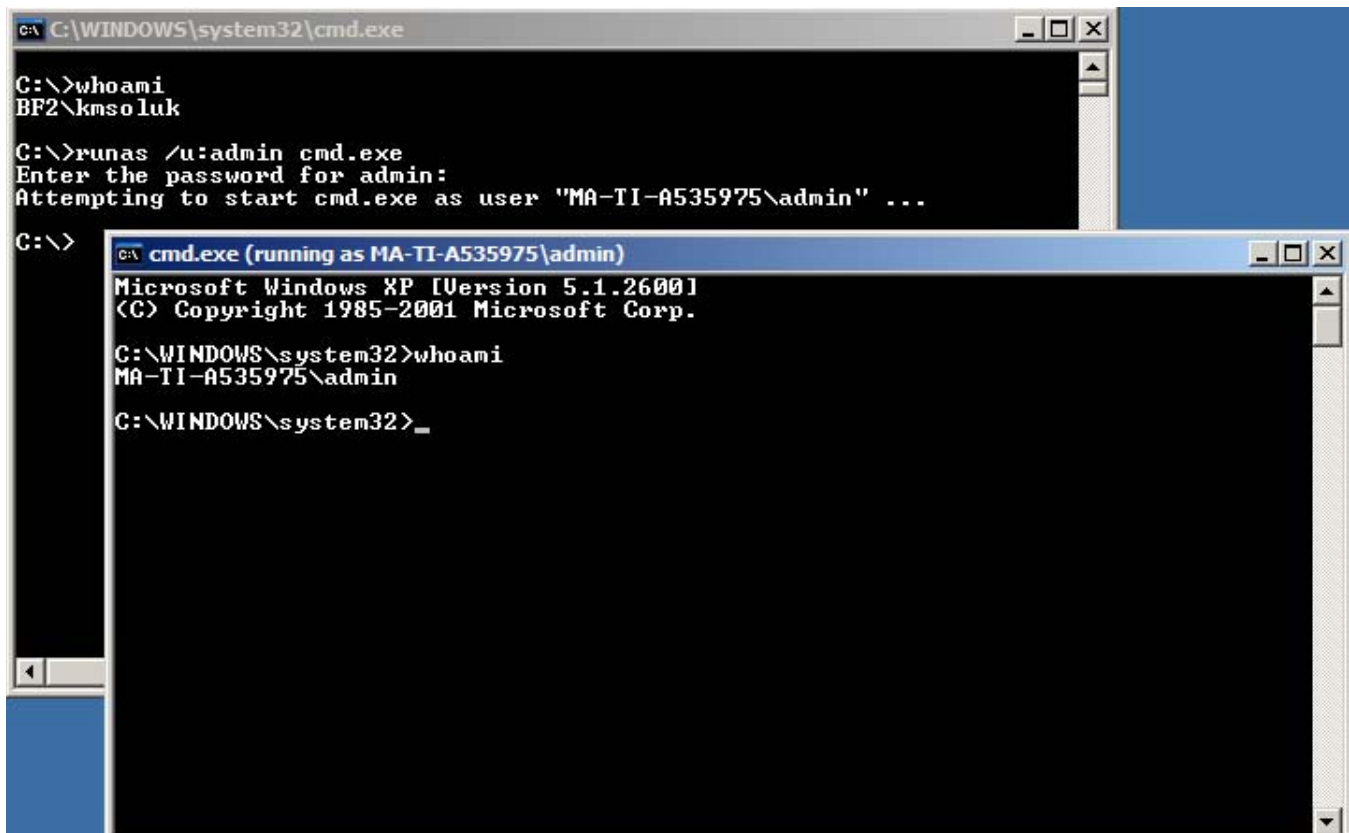
In this section, we'll describe the concepts, features, and tools you'll need to run as user and still maintain your sanity.

RunAs

RunAs allows you to launch a process under a user context that is different from the one you are currently logged in with. For example, if you are currently logged in with a domain account, you can use RunAs to open up a command shell under the context of a local admin account. Subsequent programs launched from that command shell will also run as admin. RunAs is built in to XP and manifests itself in two ways: As a command line utility and through integration with the Windows shell.

RunAs Command Line Utility

The following screenshot shows a domain user using runas.exe to launch a command shell under a local administrative context. Note the results of the whoami command in each window as well as the change in the title bar. Anything launched from the second command shell will also inherit the local admin context:



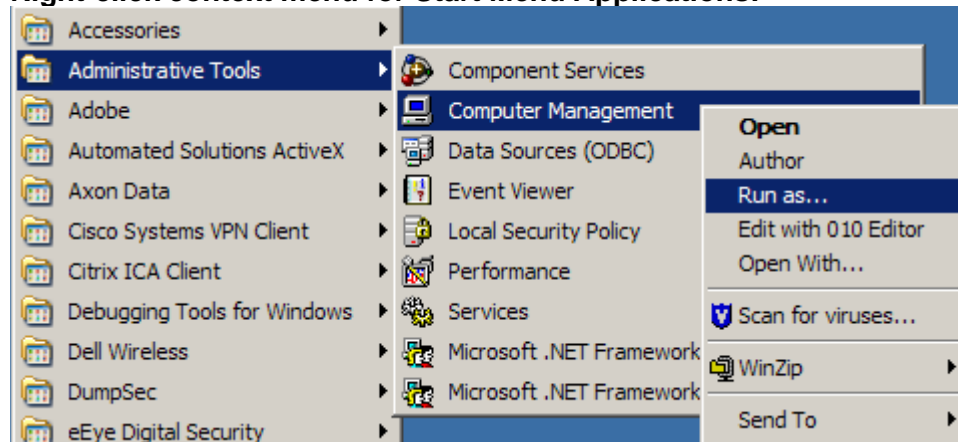
```
C:\WINDOWS\system32\cmd.exe
C:\>whoami
BF2\kmsoluk
C:\>runas /u:admin cmd.exe
Enter the password for admin:
Attempting to start cmd.exe as user "MA-TI-A535975\admin" ...
C:\>
```

```
C:\WINDOWS\system32\cmd.exe (running as MA-TI-A535975\admin)
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\WINDOWS\system32>whoami
MA-TI-A535975\admin
C:\WINDOWS\system32>_
```

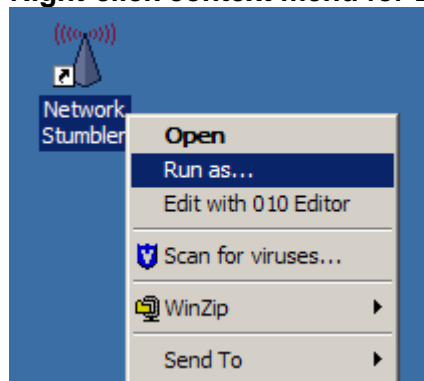
RunAs Shell Integration

RunAs is integrated into the Windows shell in several ways. The following three examples illustrate how to use RunAs to launch administrative applications from the Windows UI:

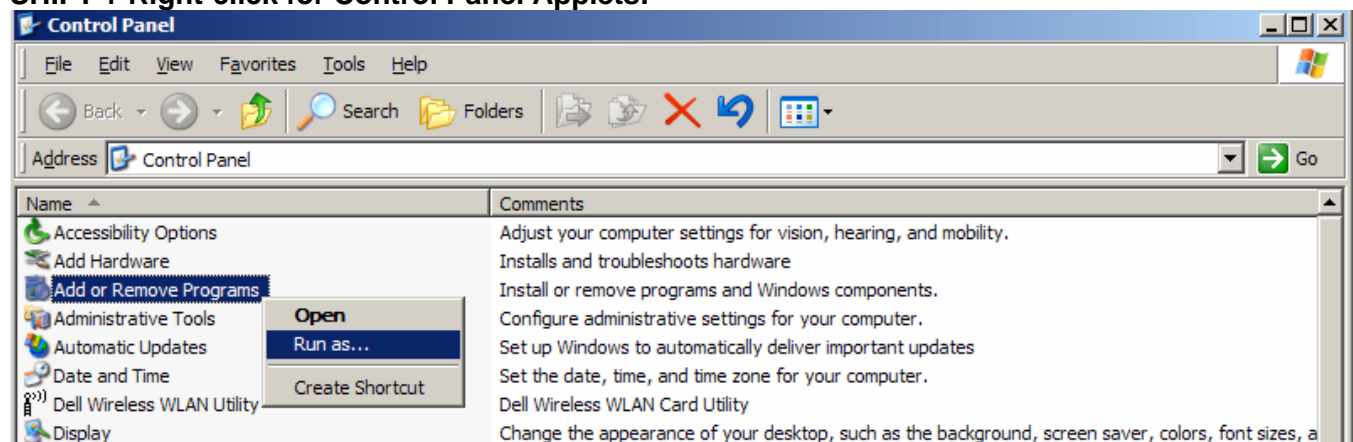
Right-click context menu for Start Menu Applications:



Right-click context menu for Desktop Shortcuts:

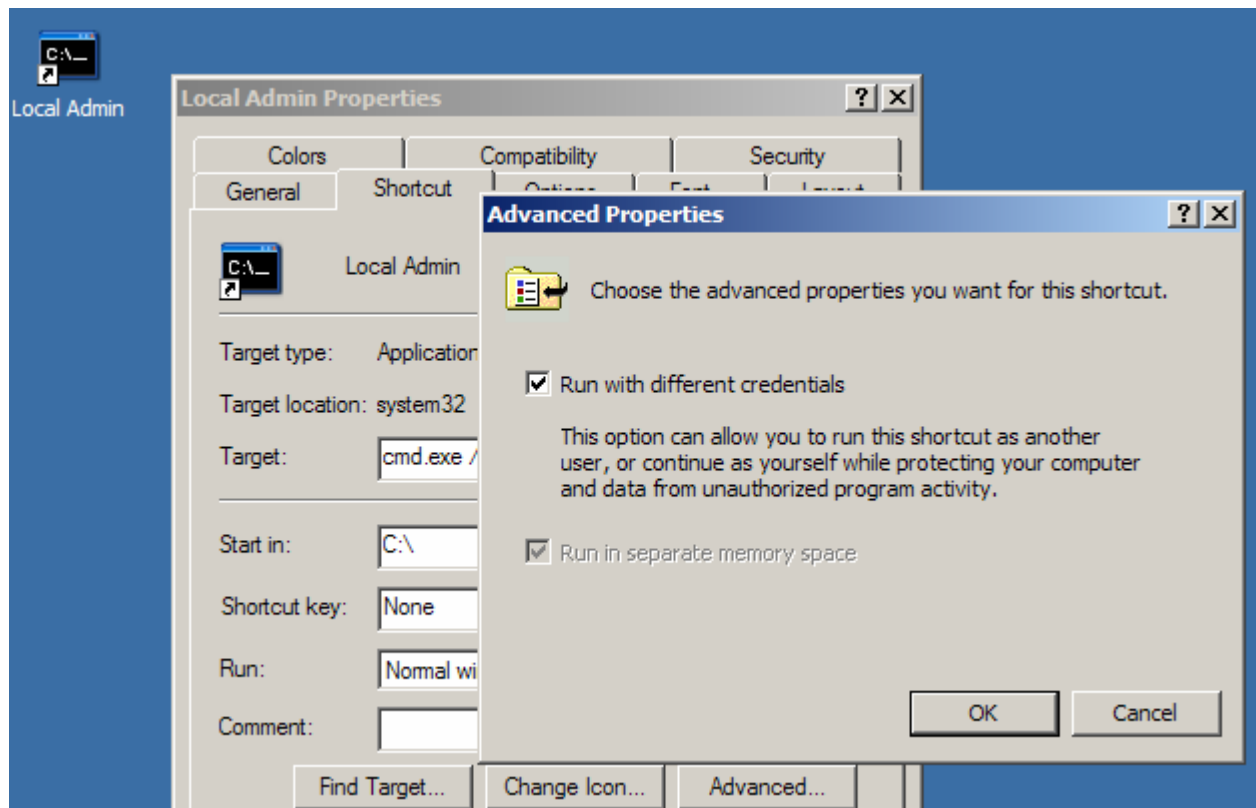


SHIFT + Right-click for Control Panel Applets:

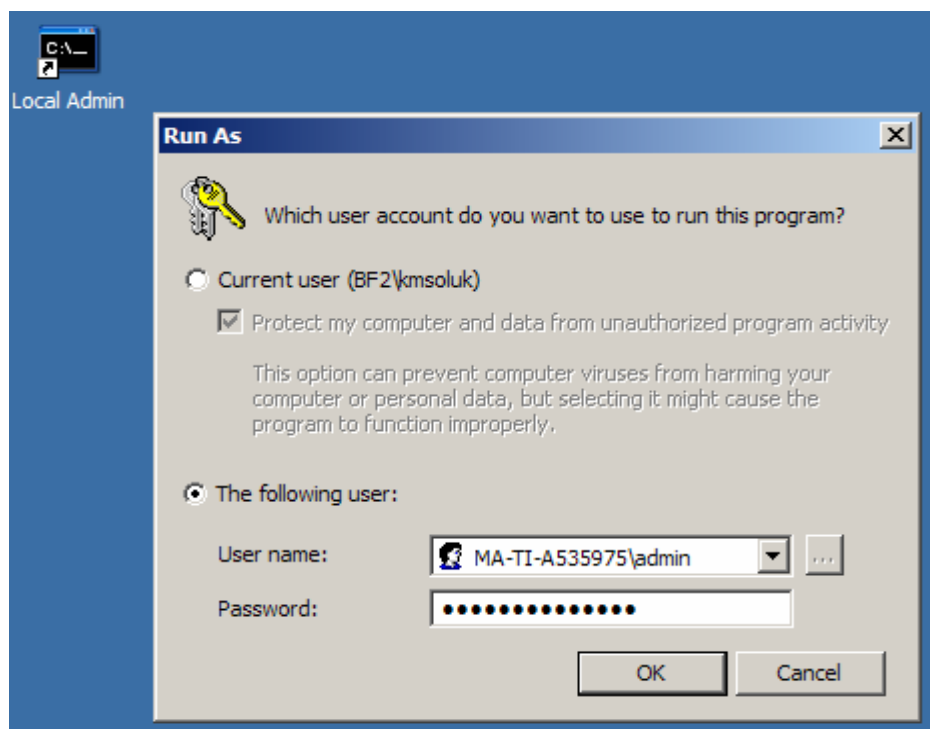


An Advanced Property for Shortcuts

When you know a certain application is likely to be run under a different context, you can set the “RunAs property” for the shortcut:

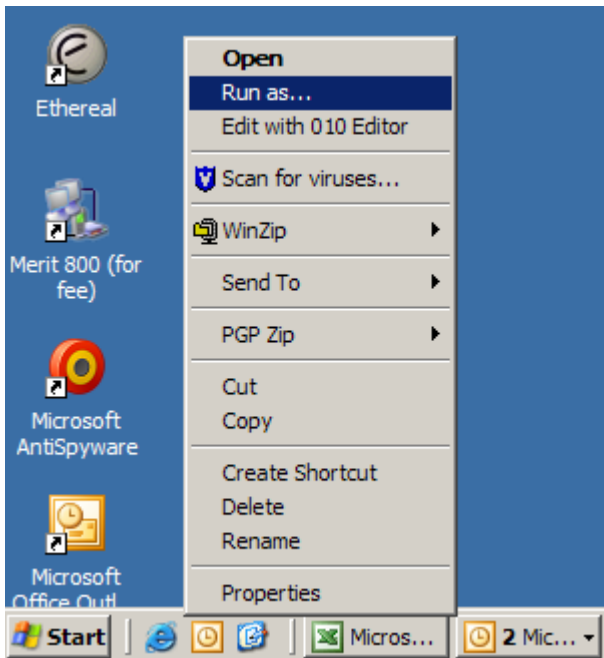


When the short-cut is launched, the shell automatically prompts for credentials:



RunAs and Internet Explorer

RunAs is not available from IE's desktop shortcut. Instead, you need to right-click on IE from the Quick Launch toolbar:



If you've removed IE from the Quick Launch toolbar, create a new desktop shortcut with the following target:

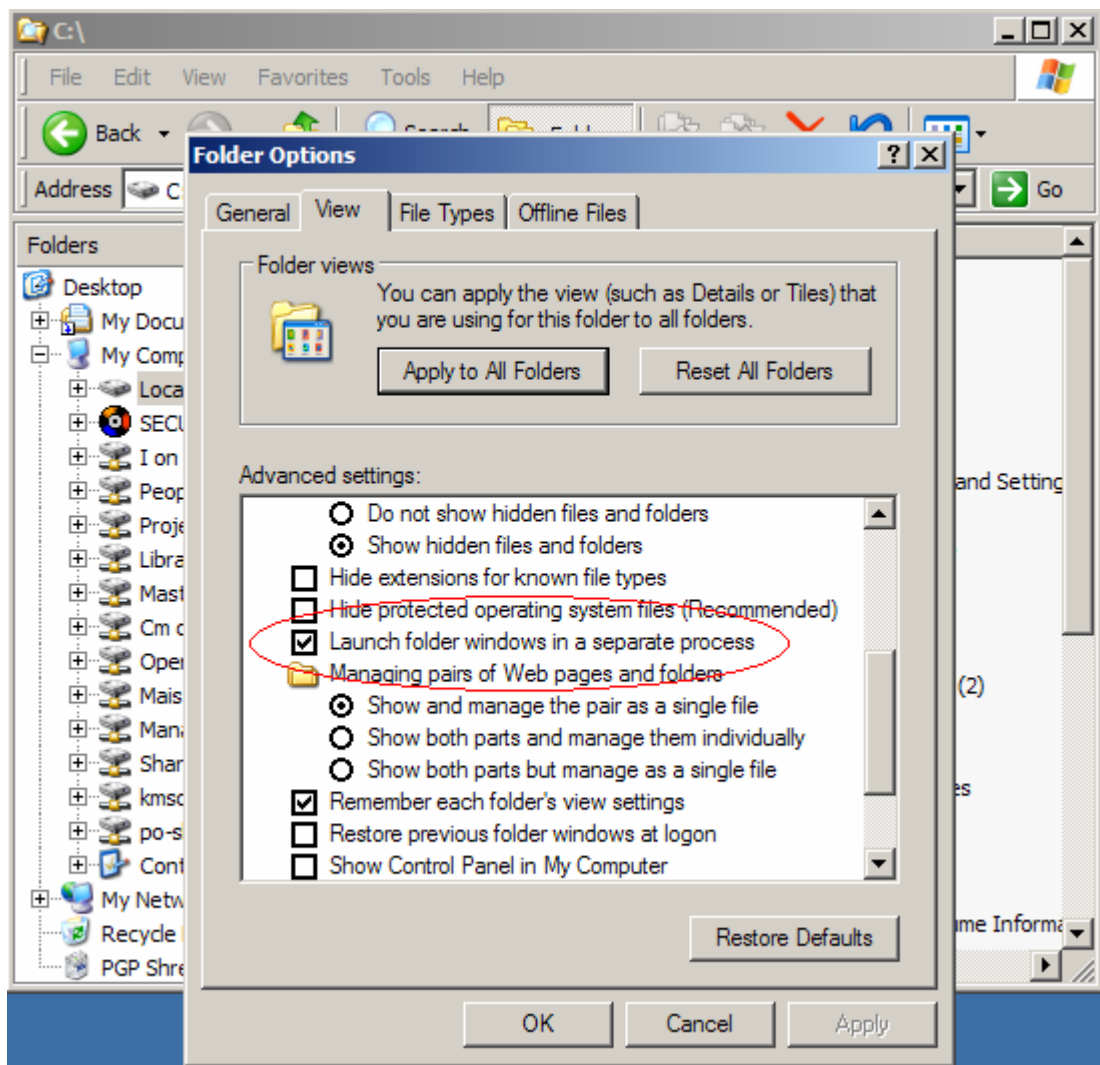
```
"C:\Program Files\Internet Explorer\IEXPLORE.EXE"
```

Optionally, drag and drop the new shortcut onto the Quick Launch toolbar. Once you have the shortcut, you can Right-click on it and use RunAs regardless of whether it's on the desktop or on the Quick Launch toolbar.

RunAs and Windows Explorer

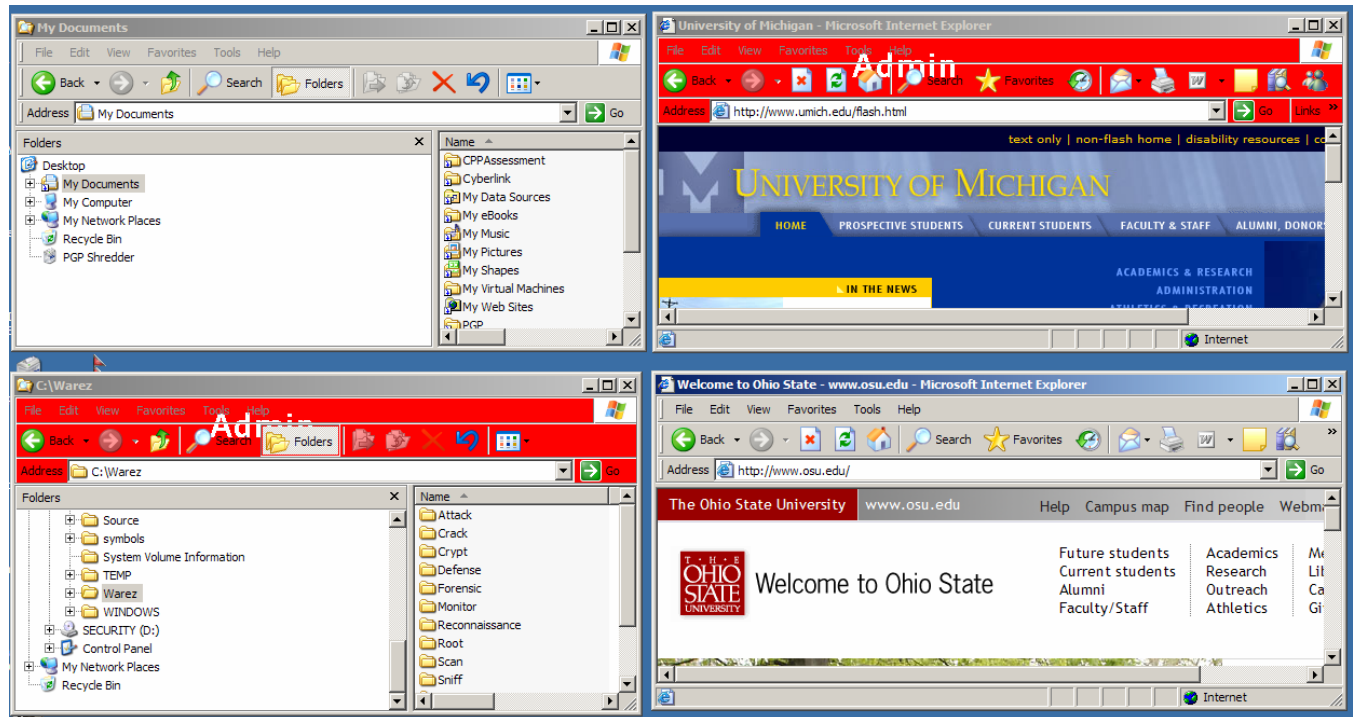
By default, Windows Explorer (i.e. explorer.exe) reuses instances of itself. In other words, by default, there is only one instance of explorer.exe running regardless of how many distinct explorer Windows you have open. This is a problem because an instance of explorer.exe is spawned as soon as you log in. After all, explorer.exe is the Windows shell! Thus, after you log in and use RunAs to launch Windows explorer, any alternate credentials you supply are moot. The new explorer Window simply runs as a thread within the initial explorer.exe shell and inherits the security context of your console logon.

To address this problem, we need to get subsequent instances of Windows explorer to launch in a separate process space. Fortunately, there is an option for precisely this functionality. From the Tools Menu in Windows Explorer, select Folder Options, then View. Check the option to *Launch folder windows in a separate process*:



Visual Distinction for Explorer Windows

Now that we can run IE and Windows Explorer in different contexts on the same desktop, it will be critical to distinguish admin from non-admin Windows. If you are viewing this document in color, the following screenshot illustrates how useful this can be:



In the next (step-by-step) section entitled *Setting up your Environment*, we'll show you how to change the toolbar backgrounds in IE and Windows explorer to achieve this visual distinction between admin and non-admin explorer windows.

Setting up your Environment

In the end, “Run As User” means logging in as a normal (non-admin) user then elevating to admin only when necessary and without having to log out. The tools, concepts, and features described above lay the groundwork for making “Run As User” feasible. In this section, we’ll provide step-by-step instructions for setting up your machine to Run As User.

Assumptions

- You are logged in to an XP/SP2 workstation with a domain account
- Your domain account is a member of the local administrators group

1. Create a local administrative account

Please note the following before proceeding with steps **1. A.** and **1. B.** below:

- You do not need to perform these steps if you already have a local administrative account and password.
- Inform your IT staff prior to creating or changing any local system accounts.
- The following steps create an account named “admin”. However, you can use any account name you wish.
- Use a strong password for this account. Either a long (greater than 15 character) pass **phrase** that is easy to remember or a shorter complex password that you can’t remember and need to keep in your wallet or store in a utility like password safe (<http://passwordsafe.sourceforge.net/>).

1. A. Create a local user account

```
C:\>net user admin * /ADD
Type a password for the user:
Retype the password to confirm:
The command completed successfully.
```

1. B. Add the newly created local account to the local administrators group

```
C:\>net localgroup administrators admin /add
The command completed successfully.
```

If you are only going to use this account interactively, modify the local security policy so the new local admin account cannot be used to access your machine over the network.

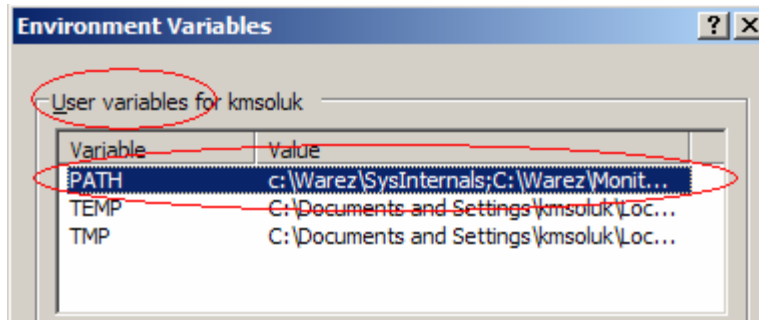
- Start → Run → secpol.msc
- Local Policies\User Rights Assignment
- Appropriately modify either of the following rights (if your domain policy allows)
 - *Access this computer from the network.*
 - *Deny access to this computer from the network*

2. Configure local admin preferences

2. A. Note your current per-user PATH settings (optional)

You may have user-specific paths set up to certain administrative applications that you will now be running under a different user context. If this is the case, you'll need to set the per-user PATH environment variable for the newly created admin. You can identify the current per-user path settings for your domain account as follows:

- Right click on My Computer → Properties
- Select the Advanced Tab
- Click the Environment Variables Tab



2. B. Log out and log back in with the newly created local administrative account

- Remember to specify "(this computer)" as the login domain.

2. C. Configure Windows Explorer to launch in a separate process

Use regedit.exe to set the following registry value to 1:

KEY: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
VALUE: SeparateProcess (DWORD)

2. D. Configure toolbar backgrounds for IE and Windows Explorer

2.D.1 Download the following bitmap into your Windows directory

<http://www.itss.umich.edu/tools/download/admin.bmp>

- Type the above URL into IE
- Right click on the bitmap - Save Picture As
- C:\Windows\Admin.bmp

There is nothing special about this bitmap. You can modify it or create your own.

2.D.2 Configure IE and Windows Explorer to use the downloaded bitmap

Use regedit.exe to create and define the following string value:

KEY: HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar

VALUE (REG_SZ): BackBitmap

DATA (Path to downloaded bitmap): C:\Windows\admin.bmp

2. E. Configure per-user PATH settings for the new Admin account (optional)

If your domain account had per-user PATH settings to administrative applications (see step 2. A. above), then define the same per-user PATH settings for the new admin account.

3. Configure your domain account environment

3. A Log out and log back in with your domain account

3. B Create a shortcut to launch an Administrative console

3.B.1 Create the initial shortcut

- Right-click on the Desktop → New → Shortcut
- **Location:** %windir%\system32\cmd.exe /t:fc /k title !!!!! Local Admin Console !!!!!
- **Title:** Admin Console
- **Finish**

3.B.2 Change the properties of the shortcut

- Right-click on the shortcut just created → Properties
- Advanced
- Check the box to "Run with different credentials"
- OK out

3. C Change the "Start-In" directory (as necessary) for existing shortcuts

Since you are joined to a domain, you will likely have several shortcuts that are set to "Start-in" %HomeDir%%HomePath%. The problem is that these environment variables have no meaning for your local admin account. You will need to change any such "Start-in" location for any shortcut that you want to launch under different credentials:

- Right-click the IE icon on the QUICK LAUNCH toolbar → Properties
If the "Start in:" location is set to %HomeDir%%Homepath% (or something similar), change it to %windir% e.g.
- Right-click your Windows Explorer shortcut or Start Menu item → Properties
If the "Start in:" location is set to %HomeDir%%Homepath% (or something similar), change it to %windir% e.g.

3. D Re-ACL non-profile data directories

If you create and store files on your local system outside of your profile directory², you will likely need to grant yourself explicit access to those files. The reason for this is that (by default for domain-joined XP systems) when you create a file as a member of the administrators group, *you* are not considered the *owner* of the file. Instead, the *administrators* group is considered the owner. Since permissions are often granted based on ownership (via the "well-known" Creator/Owner SID), you will no longer have access to such files after you are removed from the local administrators group.

To address this scenario, review the permissions on non-profile directories where you create and store files. For example, if you have a directory called C:\Data where you create and edit files, the default permissions on this directory will be:

² By default, your profile directory is rooted at c:\documents and settings\<<your user name>

- Administrators: Full Control (This Folder, Subfolders, and Files)
- System: Full Control (This Folder, Subfolders, and Files)
- Creator Owner: Full Control (Subfolder and Files)
- Users: Read (This Folder, Subfolder and Files)
- Users: Create (This Folder and Subfolders)

Thus, after you remove your domain account from the local Administrators group, you will no longer be able to modify existing files (you'll just be able to read them).

This scenario does not apply to files stored in your user profile because, by default, your domain account is granted explicit access to your profile directory. Similarly, you should not have to worry about permissions on network shares because removing your domain account from the *local* administrators group has no impact on remote machines.

4. Make yourself a non-admin

Now for the moment of truth:

4. A. Confirm that you know the password for a local admin account

4. B. Remove your domain account from the local administrators group:

```
C:\>net localgroup administrators YourDomain\YourAccount /del
The command completed successfully.
```

4.C. Log Out

Until you log out, the local administrators group will remain a part of your access token.

Run as User Walkthrough

The following “walkthrough” will help you test and familiarize yourself with your new (more secure) environment. Specifically how to switch between admin and non-admin contexts on the same desktop.

1. Log-in with your domain account

2. Verify you are no longer a member of the administrators group:

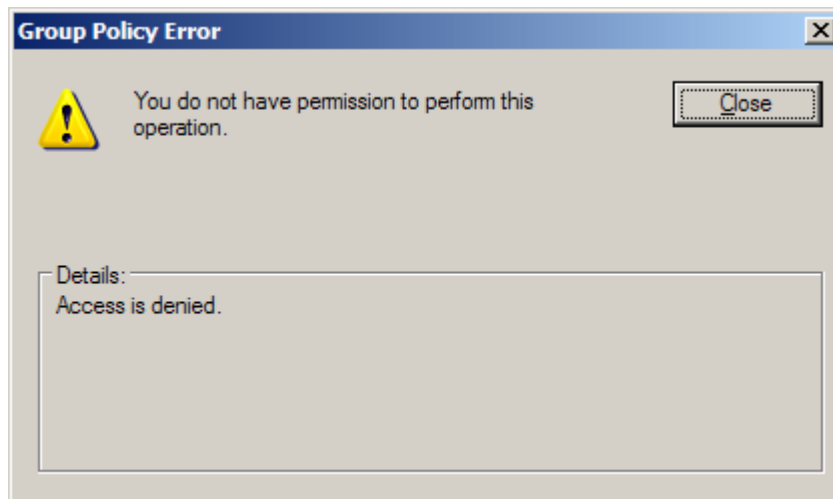
```
C:\WINDOWS>net localgroup administrators
```

You should no longer see your domain account in the members list.

3. Try to run an administrative tool:

```
Start → Run → SecPol.msc
```

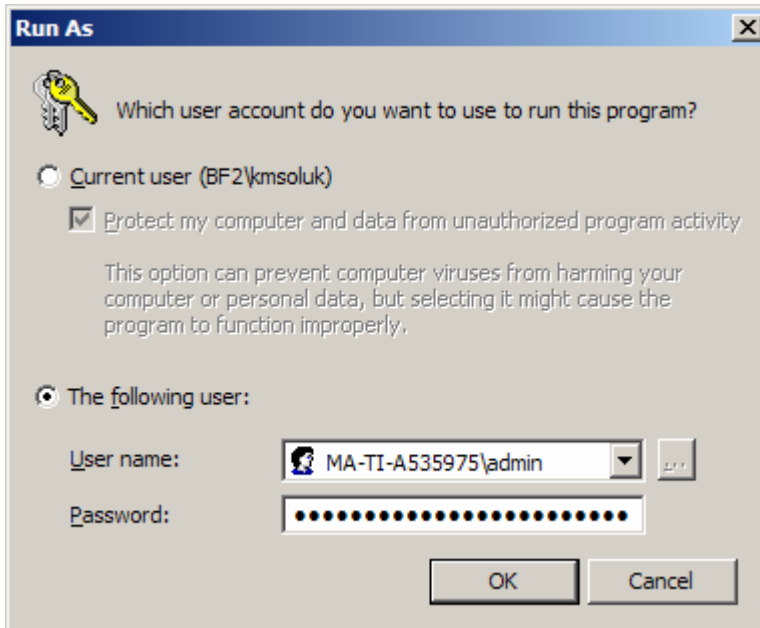
You should receive an access denied message as you are no longer an administrator and, therefore, cannot carry out the tasks provided by the tool:



4. Launch an Administrative Console:

4.1 Double-click on the desktop shortcut labeled “Admin Console”

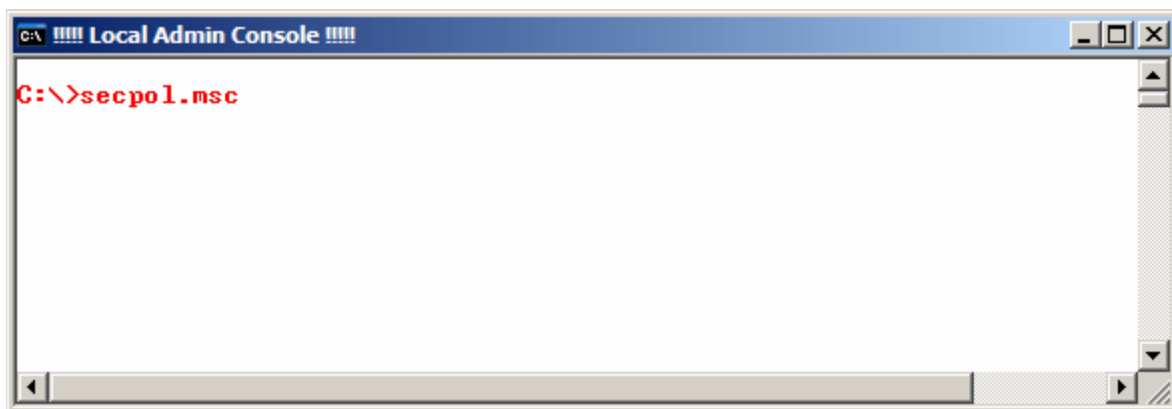
You created this shortcut in step 3.B. above. You should be prompted for alternate credentials:



4.2 Provide the username and password for a local administrator account

- Select the option labeled: *The following user:*
- Select the down-arrow for a list of the local administrators including the account you created in step 1 above.

Upon clicking OK, you should be presented with a visually acute command window:



The title bar, foreground and background colors are a result of the command line you specified in step 3.B.1 above (re-iterated here for convenience):

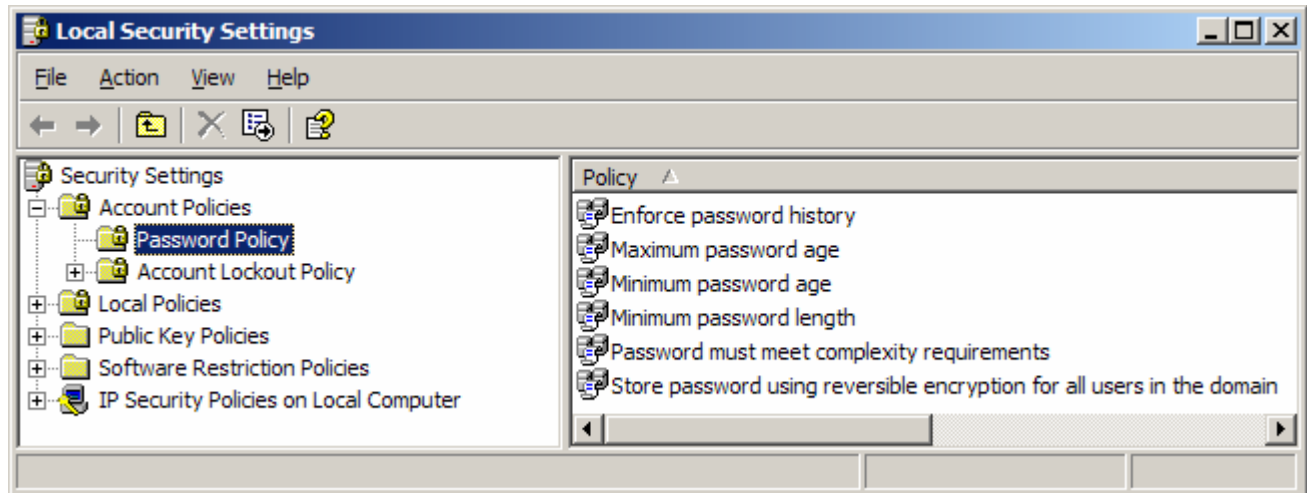
```
%windir%\system32\cmd.exe /t:fc /k title !!!! Local Admin Console !!!!
```

5. Successfully run the administrative tool that previously failed:

From within the administrative console window:

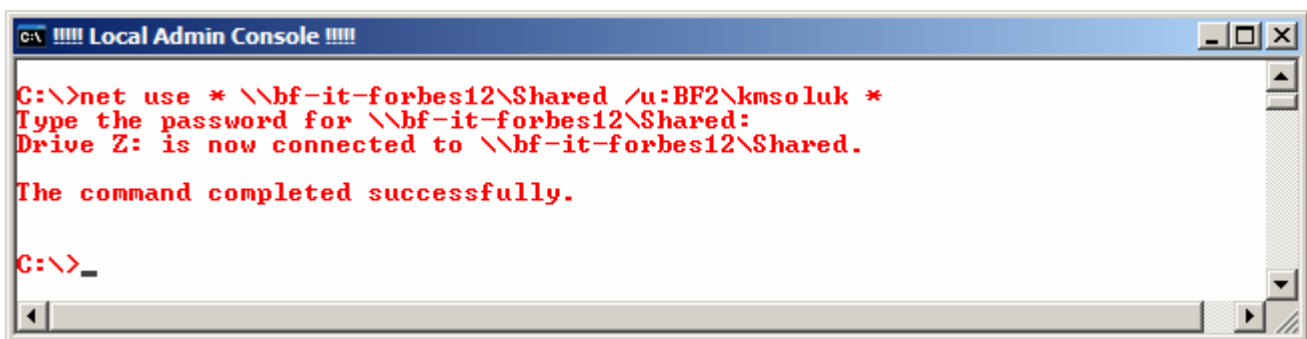
```
C:\>secpol.msc
```

This time, instead of the access denied message, you should be viewing the local security policy:



6. Connect to a network share using your domain credentials

You will need to do this, for example, to install applications that are hosted on a network share. You need to be an administrator locally to install an application, but the local administrator account doesn't have permissions to the network share. So you use your domain credentials to connect to the network share from within the administrative console:



Your local security context is that of the administrator (as visually indicated by the title bar and red font within the console window). Your security context on the remote file server is that of your domain account. This combination allows you to read (run) a setup program from the remote file server that modifies your system wide resources locally as required by most install programs.

7. Run Internet Explorer as a User and as an Admin

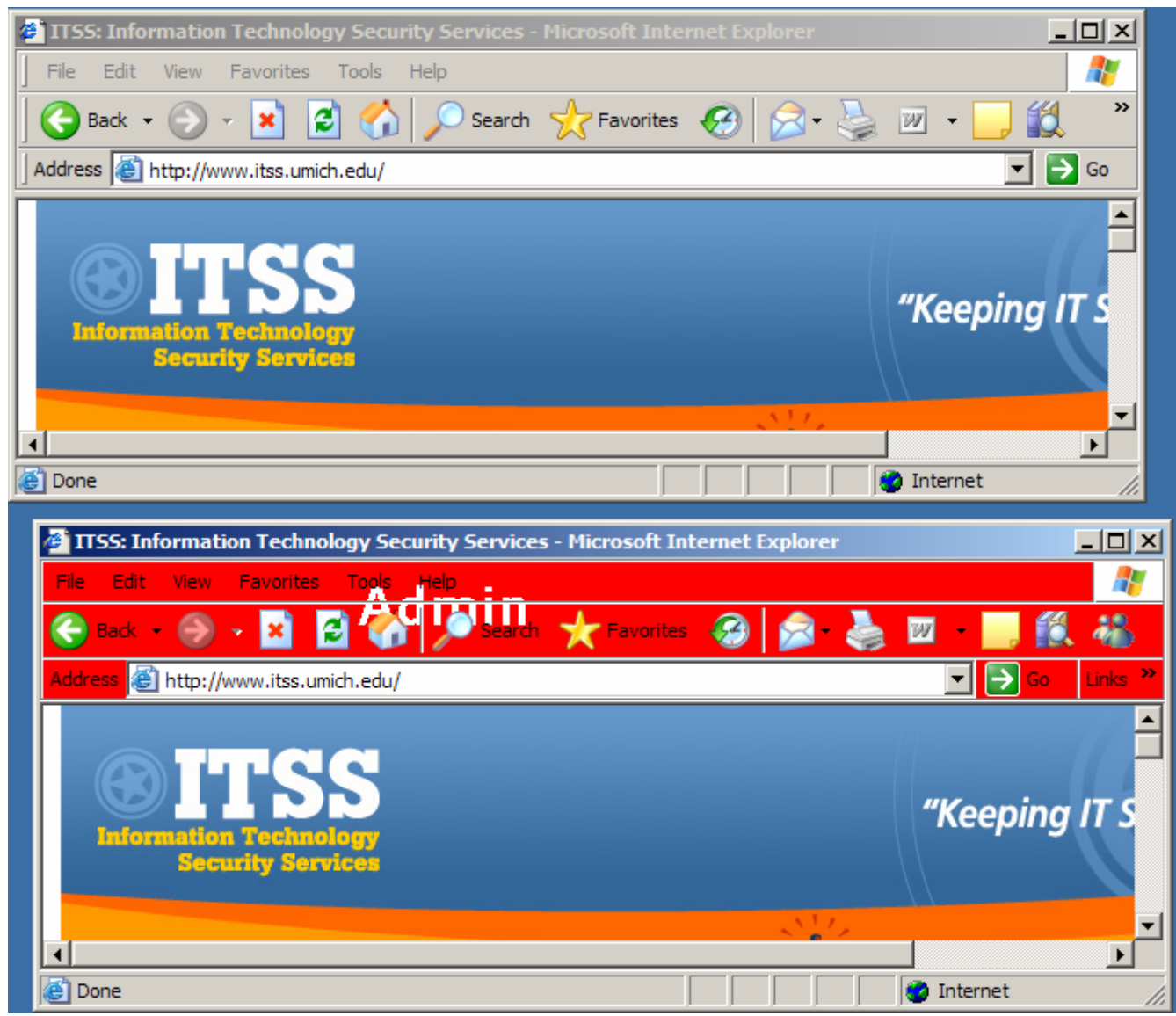
To run Internet Explorer under the more secure user context, simply launch it however you normally would. Since you are logged in with a normal (non-admin) user account, attempts to directly modify system-wide resources are denied.

To run Internet Explorer as an admin, e.g. to install trusted ActiveX controls:

- Right-Click IE on the Quick-Launch toolbar.
- RunAs

If you've removed IE from the Quick Launch toolbar, create a new desktop shortcut with the following target: "C:\Program Files\Internet Explorer\IEXPLORE.EXE" and optionally, drag and drop the new shortcut onto the Quick Launch toolbar.

After you enter in the local admin credentials, you should have two visually distinct Internet Explorer Windows due to the toolbar background you specified in step 2.D above:



8. Run Windows Explorer as a User and as an Admin

To run Windows Explorer under the more secure user context, simply launch it however you normally would. Since you are logged in with a normal (non-admin) user account, attempts to directly modify system-wide resources are denied.

To run Windows Explorer as an admin, e.g. to modify file system permissions:

- Right-Click Windows Explorer from the Start Menu
- RunAs

As was the case with IE, after you enter in the local admin credentials, you should have two visually distinct Explorer Windows due to the toolbar background you specified in step 2.D above:

9. Change your Firewall Settings (or run other Control Panel Applets)

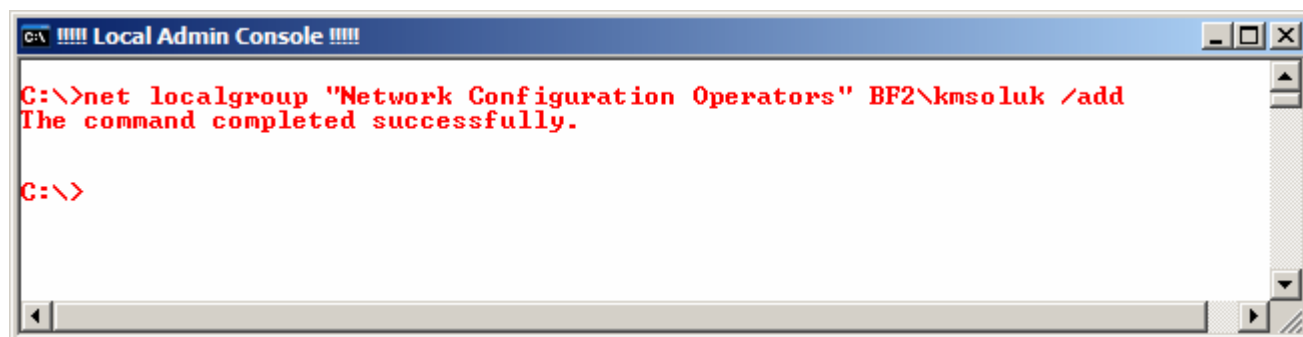
- Start → Settings → Control Panel
- Shift + Right-click on the Windows Firewall Applet
- RunAs

After you enter in the local admin credentials, you should be able to configure the per-machine firewall settings as necessary.

RunAs Gotchas

1. Network Control Panel Applet

RunAs is not available by Shift + Right-clicking on the Network Connections control panel applet. This is because that CPA is simply a wrapper that launches the Network Connections program via *ShellExecute* (which thus, runs under the context of the shell). If you need to frequently change network configuration information (a common requirement for consultants that roam around to different networks), add your domain account to the Network Configuration Operators group:



```
C:\> net localgroup "Network Configuration Operators" BF2\kmsoluk /add
The command completed successfully.

C:\>
```

Note: You will need to log out and log back in before this change is recognized.

This way, you'll be able to modify network configuration information directly using your domain account rather than having to log out and log-in as admin every time you need to change your network configuration.

2. Ctrl+C does not work in the Admin Console

This "issue" has been known since before Windows 2000 was released and has finally been fixed on Windows Server 2003. For XP, you need to use CTRL+Break instead.

3. Automatic Refresh in Windows Explorer

When running Windows Explorer on the same desktop under alternate credentials, automatic refresh does not work. For example, if you delete a file, the file will remain visible in the explorer window until you manually refresh (e.g. by hitting F5).

4. Some Resolvable Application Issues

Some applications will run successfully under a user context but take a little "tweaking" to get there. Two common scenarios are:

4.1. Missing Shortcuts

Some applications create start menu or desktop shortcuts only for the user that installed the application. When you install such an application as a local administrator, then try to run the application under your non-admin domain account there is no shortcut. This specific issue is easily addressed by copying the shortcut from the local administrators profile to either the "All Users" profile or your user-specific profile.

4.2. First run requirement

Some applications perform per-machine "initialization" steps either the first time the application is launched or each time a new user launches the application. Since system-wide changes are not allowed under a non-admin context, these application fail when you first run them under your non-admin domain account. Ultimately, you may be able to run such applications under a non-admin context by

- Performing the first run of the application under the local admin context, or by
- Making yourself an admin the first time you run the application

Once the one-time initialization activities have been performed, you can then run the application successfully under a (non-admin) user context.

5. No Access to Domain Resources

For poorly written (non-administrative) applications that simply will not run successfully under a user context you can use RunAs to launch them under an admin context. The problem then is that the local admin account does not have access to domain resources. If you need to access SMB-based domain resources from within such an application, your best bet is to launch the admin console window then, from within the admin console window, establish SMB connections to the target resources using your

domain account credentials. After you've established the SMB connections launch the application. See step 6 in the "walkthrough" above for a demonstration along these lines.

6. MSI files

RunAs is not available from the right-click context menu for Microsoft Installer applications (i.e. .msi files). To address this, simply launch the MSI from the admin console window. See also gotcha #5 in case the .msi installer application needs to access domain resources.

7. Windows Update

The web-based interactive version of Windows Update does not work when accessing it from IE under a local admin context via run-as. This should not be an issue for domain-joined (i.e. managed) machines which typically receive their patches from either Systems Management Server (SMS) or via Automatic updates running against Windows Update or against a local Windows Update Server.

8. Power Options

Changing your power configuration requires write access to per-machine locations in addition to HKCU. In order to configure your power options, use Regedit.exe to give yourself (or Interactive)

- Set Value, and
- Create Subkey

permissions on the following two registry keys:

- GlobalPowerPolicies
- PowerPolicies

Both keys are located under:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Controls Folder\PowerCfg\
```

Troubleshooting

**1. You get the following confusing pop-up message when launching a program as admin:
[c:\windows\filename.exe The directory name is invalid.]**

Assuming the directory name is indeed valid, the problem is most likely with the "Start-in" property for the shortcut used to launch the application. Right click the shortcut and select properties. Verify the "Start-in" directory is appropriate for the local administrator account. See step 3.C above for further information.

Warning and Conclusion

Running as user does not add any additional protection insofar as direct access to per-user data is concerned. When you are running as user, any piece of code that you launch can access your personal files with the same level of permission that you have. Fortunately, most malware (currently) makes the same invalid assumptions as the poorly written applications that cause you to run as admin in the first place. For example, review Symantec's *technical details* regarding the Bagle/Beagle worm:

<http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.av@mm.html>. Here is a list of some of the things this worm does:

- Writes files into the windows directory
- Terminates process (such as Anti-virus software)
- Stops and Disables Internet Connection Sharing and Microsoft Security Center
- Sets up backdoors on TCP port 81
- Modifies HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

All of these operations fail when running as user. Thus, while running as user is by no means a “silver bullet”, running as user is a security best practice and does offer significant protection against the techniques that characterize today's malware.

References

[1] Rita C. Summers, *Secure Computing: Threats and Safeguards*, McGraw-Hill, 1997

[2] Aaron Margosis' Web Log, http://blogs.msdn.com/aaron_margosis/archive/2004/06/17/157962.aspx

Resources

- Non-Admin blog: <http://blogs.msdn.com/aaron%5Fmargosis/>
- Non-Admin Wiki: <http://nonadmin.editme.com>
- “Browsing the Web and Reading E-mail Safely as an Administrator”:
<http://msdn.microsoft.com/library/en-us/dncode/html/secure11152004.asp>
- SysInternals Tools: <http://www.sysinternals.com>
Tools like process explorer, regmon and filemon are invaluable when troubleshooting “RunAs User” issues.

Appendix A: Run as User for Non-Domain joined machines.

Compared to the domain-joined scenario described in this document, it's actually easier to set up the "Run as User" environment on Windows XP Home machines or Windows XP Professional machines that are not joined to a domain. This is due to the "Fast User Switching" (FUS) feature available with XP Home and unjoined XP Pro. FUS allows more than one user to be logged on simultaneously. Thus, you can create one local administrator account to perform administrative tasks and to run poorly written (non-administrative) applications and you can make all of your other accounts non-admin users. To switch between contexts, use Window Key + L. When you move from your user account to your admin account or vice versa, all of your applications continue to run. Here are some additional pointers:

- The local admin account does not necessarily need a password. By default, accounts with no passwords can only access the machine interactively. Accounts with blank passwords cannot access the machine over the network. This is configured via the local security policy option:

Accounts: Limit local account use of blank passwords to console logon only

- Set up a different desktop background for you admin desktop. You'll want to visually distinguish your local admin desktop from your normal (non-admin) user account desktops. For example, change your desktop background color and add some text that indicates you're logged in as admin.

Appendix B: Running as User on Windows Server 2003 systems

Compared to the Windows XP-based scenario described in this document, it's actually easier to set up the "Run as User" environment on Windows Server machines. This is because Windows Servers support Terminal Services for Remote Desktop Administration. Thus, while you are logged in with your non-admin account, you can "terminal server" back into your machine and log in with administrative credentials. For example, create a shortcut on your non-admin desktop with the following target:

```
%windir%\system32\mstsc.exe /v:127.0.0.1
```

As is the case with "Fast User Switching" described previously in Appendix A, the terminal services solution allows two simultaneous logons and the ability to toggle back and forth between them. Thus, it is a good idea to visually distinguish your admin logon in this scenario as well.