

XP Professional Security Features: An Introduction

Tim Mullen 2002-06-05

XP Professional Security Features: An Introduction

by Timothy M. Mullen

last updated June 5, 2002

I'm not sure if it was the uplifting backbeat of Madonna's "Ray of Light" or the promise of the best security options yet in a Microsoft operating system that made the difference, but the overall development and marketing efforts of XP have paid off. With over 32 millions copies sold since it was released last October, Microsoft's newest line of operating system products has caught the attention of both home and business users.

While XP Home has many new security features available for the home network, this article will focus on XP Professional (hereafter simply referred to as XP) and its use in the corporate LAN. This is not intended to be exhaustive dissertation of all the new features in XP; rather, the purpose is to highlight some of the new security features found in the product, and to provide those still considering an upgrade to XP with some insight into how doing so can help them administer their network. So let's jump right in.

Remote Assistance / Remote Desktop

The new Remote Assistance feature in XP is an easy way to set up a user-initiated remote control session, and is really pretty cool. Turned on by default, it offers users the capability to send a Remote Assistance request to anyone with an e-mail address or an MSN instant messaging account.

Basically, the user initiates a RA request, which sends a small file called `rcBuddy.MSRcIncident` to the recipient. This is really just a little XML file that contains the connection information it needs to get back to the unit requesting assistance. The file is an attachment in the recipient's e-mail, which launches the RA connection applet upon execution. Note that the remote unit must be able to connect to port 3389 on the host.

When the RA token is created, the user can choose to specify whether or not the particular connection will require a password. Of course, passwords are the preferred way of protecting the connection but, if selected, the password must be communicated outside of the request, as it is not contained in the e-mail itself. Session keys are generated to ensure that one RA token

can't be used for a different request session. Although I have not performed any cryptanalysis of the token's session key (as if I would know how to do so in the first place), it seems to be a pretty secure way of doing things. The RA token can also be given a sunset value to expire after a certain amount of time.

Even though a session token has been generated and sent, the process is still not complete. When a session connection is initiated from the remote unit, the original station receives a confirmation dialog box requiring the connection to be accepted. Once established, the remote session is in read-only screen mode, but chat between the two units is automatically enabled. If the remote unit requests remote control, the host unit must again acknowledge this request.

At this point, the caller will have remote control over the host unit in a similar manner to a PC Anywhere-type connection. NetMeeting functionality has been built in to allow voice communications over the connection as well as the aforementioned chat capabilities. At any time, the host user can terminate the session with a quick ESC.

All in all, it is a pretty functional means of giving the user the ability to request Remote Assistance, without giving them the capability of adding users for true Remote Desktop connectivity.

Remote Desktop Connection

Anyone who has been using Terminal Services in Win2k has probably already been using the Remote Desktop feature of XP. Unlike Remote Assistance where both parties see what each other are doing and can fight over the mouse and such, the Remote Desktop is simply an extension of the desktop to a remote user. And while true Terminal Services in Win2k allow simultaneous session access by a console operator and remote terminal users, XP Pro only allows one type of connection at a time. Console users will be locked out during remote desktop sessions, and remote desktop sessions will be discontinued when the user decides to log back on at the console.

But, like Terminal Services, the listening port for Remote Desktop is always on, and is subject to the same type of attacks that any persistent terminal service is prone to. Though Remote Assistance and Remote Desktop both use the core terminal services components, they may be individually enabled or disabled. In other words, you don't have to allow RA to allow RD and visa-versa. These services can also be controlled by the security policy.

The Remote Desktop Connection client is full-featured, working as a client to both XP Remote Desktop hosts and Win2k Terminal Services. Many of us know that in an attempt to obscure terminal services, we will move the listen port from 3389 to some other port unknown to potential attackers. However, I don't think many people actually do so, because the Terminal Services client requires individual connection profiles to be created and altered on a connection-by-connection basis. It's kind of a pain. But, if you use the XP Remote Desktop client, connecting to hosts with unique/changed listening ports is as easy as appending the port number to the hostname with a delimited colon as in "tshost.domain.com:12345." This simple feature really allows us to leverage yet another security-in-depth mechanism without sacrificing usability [Thanks to Jim Harrison, resident Microsoft "ISA Ninja" for that tidbit].

Terminal Services users will also have noticed that every time they connect to a server via the old TS client, the server logs several TermServDevices errors in the Application Log. Event IDs 1111, 1105, and 1106 get created every time the user connects from a client whose local machine printer drivers are not also installed on the server. It is more of a nuisance than anything else, but these events are logged as actual errors, not warnings, and that really irritates a lot of people. OK, so it probably only irritates me, but with the new XP client, you can now choose not to map remote session print spooling from the server session to the local printer, thus keeping your Application Log cleaner, and the anal log-viewing administrator happier.

Additionally, the user experience via the Remote Desktop client now includes sound feedback, better resolution and color support, and can be optimized to the available bandwidth of your connection.

Service User Contexts

XP has also introduced a couple of new security contexts in which services can now exist: Local Service, and Network Service. Local System has traditionally been used for many services running in previous versions of Win32 operating systems. The downfall here is that an attacker or trojan that controlled that service, in turn controlled the system. If the attacker gets Local System on a domain controller, then the entire domain is hosed. Sometimes, these services need access to network resources - domain user or delegated accounts were typically used in these cases. But again, if control of this type of service was seized, other network resources down-range could also be compromised.

Local Service and Network Service are basically glorified local user accounts, only possessing the rights of a typical local user on the system (one in the Users group). This is a much more secure context than Local System. The main difference between the two is that when a service is run in the context of the Local Service account, network resources are utilized via a null session connection. Services run under the Network Service account will connect to network resources under the actual machine account. By utilizing these account types, one can mitigate damage from a compromised service; however, be aware that many services still run under Local System by default.

Internet Connection Firewall

XP is the first Microsoft OS to offer built-in personal firewall capabilities. The Internet Connection Firewall, or ICF, is a happ-nin' little app that is automatically enabled on Internet connections created via the Network Connection Wizard, and can be manually enabled on any network connection. It is secure in its default configuration, and can be customized to allow specific TCP or UDP ports at the connection profile level. One of the main criticisms I have heard regarding the ICF is that it does not perform packet filtering at the egress; but as many people seem to agree, a malicious process generating packets internally could own the machine anyway, which means it could own the ICF as well, and simply turn it off. Many believe the value of personal firewalls lives in its ability to drop packets at the ingress.

Over that last several months, much vitriolic diatribe has been published regarding the evils of Raw Sockets and XP's support of them. Of course, many know that that this support was built into Windows 2000 and has been available to any version of Windows before that through packet drivers. People have pounded their chests and roared about the pending crippling of the Internet via malicious use of raw sockets in XP, saying that spoofed IP attacks would emanate from all the unsuspecting XP users on the Internet. However, no one seems to bother to mention that XP's Internet Connection Firewall, which is turned on by default on Internet connections, actually prevents spoofed packet headers from exiting the interface.

As you will learn in the soon-to-be-published XP Security book from Foundstone (published by Osborne), XP's ICF examines the headers of outbound packets, and will reconstruct them to ensure that an incorrect source address can't be programmatically altered in the stack, not to be able to go anywhere, that is. This won't prevent DoS attacks, but it will prevent spoofed DoS attacks, which is what the alarmists seem to be up in arms about. But again, since DoS trojans had to get on the machine in the first place, an accomplished coder could check for ICF and

turn it off if he was really determined to perform a spoofed attack.

File and Settings Transfer Wizard

Though not really a security feature of XP, the Files and Settings Transfer Wizard (FSTW) can actually make an administrator's job easier when users are migrated to a new installation, so I thought I would take a moment for a quick overview. When moving a user to a new machine, one has to be diligent in ensuring that all user settings get properly reconfigured on the new box. Internet Favorites, e-mail accounts and client settings, screen layout and themes, start-bar configuration, and a host of other minute details need to be transferred and are easy to drop out.

I was surprised at how well this tool functioned in my tests. In a nutshell, the wizard walks you through what file types and specific directories you want to transfer. It will then identify which programs it thinks you should install on the other box prior to running the companion wizard on the target box. You have a few choices of how to perform the transfer, such as direct box-to-box, network, and shared-directory based access. When all was said and done, all of my personal preferences and settings were configured for my user account on the new machine. Settings for Windows Explorer, my desktop, Outlook, Office, and everything else were just as I had set them on the original box. What I was particularly impressed with was the inclusion of non-Microsoft software. For instance, my settings for Eudora and Winamp were completely mirrored, including screen layout, filters, email accounts, PGP settings, skins, visualizations, and play-lists.

What does pertain to security is the inclusion of the user's security and zone settings for IE. Though group policy allows the enterprise admin to automatically configure these settings, many people still do not use group policy in these cases; as a result new installations with the default zone settings can be dangerous. The FSTW ensures that the original security settings are transferred over to the new user's profile.

Not everything was perfect, though. Passwords for dial-up accounts or VPN connections were not transferred (probably by design), and it looked like the Domain Name on these connections was truncated in several instances. I was also concerned when I saw the FSTW fail to transfer Internet Connection Firewall settings from old profiles to the new ones, leaving previously ICF protected connections without any firewall settings. If you have connections using ICF, be sure to double check the settings on the new box after the wizard has completed.

And finally, in my tests, files protected by EFS encryption were not transferred, even when the source and destination boxes were logged in as the same domain user.

Even given the issues I had with the tool, it is still a valuable addition to the OS - just make sure you give it a good once over before you get used to the install-transfer-nuke process.

Local Security Policy

While there are still many other powerful security features in XP, we'll end this segment with the new options in the Local Security Policy. Expanding on the functionality created in Window2000 Group and Local Security Policy, XP now has some new security settings available that you should be aware of.

I think the first thing to talk about is XP's default setting to limit remote interactive (non-console) logons for accounts without a password. If you create an account without a password, that account can't be used for network-based connections to the machine. Though not a cure-all in that console logons and domain user accounts are still not affected, it is strong setting for Microsoft to choose to enable by default. You will find this setting under the "Local Security Settings/Local Policies/Security Options/Accounts: Limit local account use of blank passwords to console logon only" node.

Null Session Settings

XP has also introduced many new Anonymous (null) session restrictions. Even in Win2k, one could typically retrieve an abundance of information regarding user accounts from a domain controller, even when explicitly restricting anonymous connections (when setting RestrictAnonymous to 1; Win2k's support of a value of 2 killed you). XP now supports a number of very specific restrictions for null connections:

- **Network access: Allow anonymous SID/Name translation**

Anonymous users have always been able to retrieve the SID for any account by providing the account name, or the account name by providing the SID. The combination of persistent user/ group names such as "Guest" or "Domain Users" would allow a connection to retrieve the domain's SID. Using persistence RID ID's (such as 500 for administrator) or SID-walking programs, (like [UserDump](#) written by some dude named

Thor) an anonymous connection could retrieve all the user names in a domain, including the administrator even if it was renamed. This setting finally limits the ability to do so.

- **Network access: Do not allow anonymous enumeration of SAM accounts**

This setting will keep anonymous users from getting to the SAM accounts. Honestly, this is not that big of a deal for XP workstations, but it will be a valuable setting for .Net clients with various domain trusts who want to limit access to null users.

- **Network access: Do not allow anonymous enumeration of SAM accounts and shares**

Same as above, but with the additional restriction of share information.

- **Network access: Let Everyone permissions apply to anonymous users**

XP now removes the 'Everyone' group from the token generated for anonymous tokens by default. This was not done previously. If this behavior breaks something in your Enterprise (like SMS server discovery) then you can revert back to the legacy token with this setting.

- **Network access: Named Pipes that can be accessed anonymously**

The setting name says it all, but I couldn't just leave it without some explanation - this setting allows you to selectively determine which named pipes can be accessed with the null user, including resources like the End Point Mapper and the SQL Query pipe.

- **Network access: Shares that can be accessed anonymously**

Allows an administrator to select which network shares can be accessed anonymously.

Crypto Settings

Another new security policy in XP allows you to control encryption. **System cryptography: Use FIPS compliant algorithms for encryption** is a setting that limits TLS/SSL encryption algorithms to the `TLS_RSA_WITH_3DES_EDE_CBC_SHA` cipher. i.e., EFS by default uses DESX for encryption; enabling this setting would force it to use Triple DES.

Password and Authentication Settings

Finally, we will talk about a few settings to help you crank down on password and authentication settings. While these features have been available in Win2k, and some even in Win9x, they are now easily accessible in the Local Security Policy.

- **Network security: Do not store LAN Manager hash value on next password change**

The LM hash stored in the SAM allowed for quick and easy password cracking due to its weak hashing algorithm. If you got hold of the SAM, L0phtcrack could crack it in seconds. You may now choose to remove the LM hash altogether via this setting. One can achieve the same results for individual accounts by simply setting a password greater than 14 characters long (this works in Win2k as well).

- **Network security: Minimum session security for NTLM SSP based (including secure RPC) clients**
- **Network security: Minimum session security for NTLM SSP based (including secure RPC) servers**

These settings were also available in Win2k, as well as on Win9x machines using the Win2k client software, but are now easily configurable via the LSP. These two settings allow you to granularly control application to application security settings on clients and servers by requiring a certain level of SSP encryption. Administrators can require all or none of the following: Require message integrity, Require message confidentiality, Require NTLMv2 session security, and/or Require 128 bit encryption.

Conclusion

That pretty much does it for this article. For those of you that have stuck with me this far, I appreciate your time and hope you got some value out of the piece. Good luck with your XP deployments, and stay secure.

[Privacy Statement](#)

Copyright 2006, SecurityFocus