TechRepublic's ultimate guide to enterprise wireless LAN security

Version 1.0
January 10, 2007

By George Ou

Takeaway

Enterprise wireless LAN security is a persistent concern for every system administrator and CIO. This TechRepublic ultimate guide will give you the information you need to secure all of the wireless connections in your enterprise.

Table of Contents

Introduction to PEAP authentication

Introduction

Protected Extensible Authentication Protocol (PEAP) Authentication is a secure password-based authentication protocol created for the purpose of simplifying secure authentication. PEAP is primarily used in Wireless LAN networks though it can also be used for wired authentication, Network Access Protection (NAP), or even VPN authentication in Vista. While there are other suitable authentication protocols like Funk Software's EAP-TTLS that operate nearly identically to PEAP, PEAP enjoys native Windows operating system support along with Windows Group Policy management, which makes it extremely easy to deploy.

Why PEAP and not proprietary authentication protocols

Although PEAP has garnered a sizable market share for non-Cisco shops, it has not weaned a large portion of Cisco customers from the weaker LEAP protocol which commanded the lion's share of the Enterprise WLAN market. More recently, Cisco customers began moving to the newer Cisco EAP-FAST protocol which is almost as insecure as Cisco LEAP and a nightmare to deploy securely. What's worse is that even older Cisco wireless adapters cannot run Cisco EAP-FAST but they can all run PEAP. Because PEAP is universally compatible with virtually any hardware from any vendor, offers "machine level authentication" which is critical for the Enterprise, and can be automatically deployed in Active Directory,* PEAP is the ideal choice in authentication protocols. Note that I am not saying you shouldn't use Cisco hardware since I've had plenty of luck with Cisco's hardware reliability. I'm only recommending the use of standardized protocols for maximum flexibility, compatibility, capability, and ease of deployment.

** Only with Microsoft's built-in Windows XP/Vista PEAP client*

PEAP and PKI

Public Key Infrastructure (PKI) is a component of Public Key Cryptography (PKC) that uses Digital Certificates (x.509 format) and Certificate Authorities. PEAP uses PKI to secure user authentication from man-in-the-middle (hacker listening in the middle) attacks much the same way that SSL uses PKI to secure Websites for e-commerce or other sensitive applications.

Although PEAP and SSL operate on different layers of the OSI model (layer 2 vs. layer 5), they both use a server-side digital certificate to facilitate a secure key exchange to start a secure encryption session even if the entire session was being monitored by hostile eyes. This secure session not only protects the key exchange, but even more importantly it protects the authentication session which left unprotected may compromise the user's password.

The PKI model achieves secure key exchange by using Digital Certificates which are simply digital documents that assert their owners identity. Digital Certificates by themselves are worthless unless they are signed by a trusted entity called a Certificate Authority (CA). In order for a CA to be trusted by a client in the form of a wireless laptop using PEAP or a home computer used to shop online using SSL, a "root certificate" containing the public key of that CA must be installed in the user's Certificate Trust List (CTL).

All modern operating systems contain a preinstalled list of trusted Root Certificates in their CTL, and this is what gives a company like VeriSign the authority to sign digital certificates for servers world wide. Using a publicly trusted company like VeriSign makes PKI deployment very simple because it is already trusted by every computer or PDA device in the world off the factory floor, but the server certificate may cost hundreds of dollars per year. Using EAP-TLS with a public CA is even more costly because you would also need to shell out an additional $60 per user per year for client side digital certificates.

Private CAs allow you to sign your own digital certificates if you possessed the knowledge and the infrastructure to house your own private CA. This is why there are so many organizations that simply don't want to bother building a CA infrastructure and they don't want to spend $300 a year dealing with public CAs. This is why so many organizations choose LEAP and live under the illusion that their passwords strength alone will protect them. Convincing these customers to embrace PKI is usually an uphill battle and I speak from first hand experience. To make life easier on you, I'm going to teach you how to avoid dealing with public or private certificate authorities in this wireless LAN series by using self-signed digital certificates.

Authentication server requirements

To implement PEAP, the organization needs to implement a RADIUS Authentication Server. There are many ways to do this no matter your software preference. There are options for Microsoft Windows Server 2003 with SP1 or Windows Server 2003 R2 with IAS, 3rd party RADIUS servers such as Funk Odyssey which allows you to tie in non-Microsoft directories like Novell, and Open Source solutions like FreeRADIUS. Windows Server 2000 also had IAS but only supports EAP-TLS authentication and not PEAP authentication.

To run PEAP, the RADIUS server must have a server side x.509 digital certificate. This certificate can be purchased from a third-party Certificate Authority such as VeriSign, or it can be issued from an organization's internal Certificate Authority. These two options are conventional wisdom but neither option is particularly appealing to small businesses since they won't like paying $300/year for a third-party Digital Certificate and they probably don't have a PKI Certificate Authority server in-house. An excellent way to get around this problem is to use a Self Signed Certificate on your RADIUS server. To implement Microsoft IAS, you can follow my IAS Server configuration guide.

Hardware and Software requirements on PEAP

Every single enterprise class Access Point will support generic RADIUS authentication which is compatible with all the WPA/WPA2 certified EAP types which includes PEAP authentication. Cisco Access Points support a proprietary form of authentication that is used for proprietary Cisco LEAP and EAP-FAST protocols which isn't supported by non-Cisco access points, so I cannot recommend it. Cisco refers to their proprietary authentication method as "Network EAP" and the open authentication method as "Open EAP". As for client side hardware requirements, any client adapter that supports the Windows Wireless Client will support PEAP authentication and most Wi-Fi adapters fall in to this category.

Recent versions of Windows Mobile and CE also support PEAP authentication. Windows XP Wireless Client added PEAP support with SP1 (Service Pack 1) and enhanced it further with Service Pack 2. There was even a WPA2 update for Windows XP. Windows Vista has all the Wireless Client features.

The latest Windows 2000 Service Pack also added PEAP support but WPA grade TKIP or AES encryption capability was never added, nor can you manage Windows 2000's Wireless Client through group policy. You can get a free WPA client for Windows 2000 and many Wi-Fi adapters with Intel or Atheros come with their own Wireless Clients with WPA support for most Windows operating systems. But only the Windows XP and Vista's wireless client can be managed centrally through Active Directory Group Policy. You can learn how to manually configure Windows XP here or automatically configure Windows XP and Vista through Group Policy.

LEAP: A pending disaster in enterprise wireless LAN security

Introduction

Towards the end of year 2000, Cisco created a proprietary EAP (Extensible Authentication Protocol) protocol called LEAP (Lightweight EAP) for its line of Wireless LAN Access Points as a way to address the security weaknesses in WEP. One year after its introduction, LEAP authentication for 802.1x Wireless LAN implementations had quickly established such a strong foothold in the enterprise market that it became difficult to sell third party or integrated Wireless LAN adapters that could not run Cisco's proprietary ACU (Aironet Client Utility).

As of 2004, Cisco's commands roughly 60 percent of the enterprise Wireless LAN market and according to one survey by Nemertes, 46 percent of IT executives in the enterprise said that they used LEAP in their organizations. So what's the problem you might ask if you're not one of Cisco's competitors? Because LEAP is used by such a significant percent of enterprise Wireless LANs, it represents a massive security hole for a vast install base in enterprise Wireless LANs where security is suppose to be priority one. What's even more shocking is that few enterprises are doing anything about it.

The weakness of LEAP

The theoretical weakness of LEAP was well known from the beginning since it is essentially an enhanced version of EAP-MD5 with Dynamic Key Rotation and Mutual Authentication added. Since EAP-MD5 was never meant to be used on an un-trusted wireless medium and was only to be used for wired communications where there is at least a minimal level of physical security, LEAP probably should never have been used for Wireless LAN authentication.

LEAP is fundamentally weak because it provides zero resistance to offline dictionary attacks. This is because it solely relies on MS-CHAPv2 (Microsoft Challenge Handshake Authentication Protocol version 2) to protect the user credentials used for Wireless LAN authentication. MS-CHAPv2 is notoriously weak because it does not use a SALT in its NT hashes, uses a weak 2 byte DES key, and sends usernames in clear text. Because of this, offline dictionary and brute force attacks can be made several orders of magnitude more efficient by a very large (4 gigabytes) database of likely passwords with pre-calculated hashes. What was once thought to be a strong password just became pathetically weak and breakable within minutes.

The standard response from Cisco for years was that LEAP is secure if that organization is using complex enough passwords that make it computationally infeasible to attempt an offline dictionary or brute force attack. Although it was technically true, few organizations have password enforcement policies that meet the strict requirements of strong passwords: 10 characters long with random upper case, lower case, numeric, and special characters.

The vast majority of passwords in most organizations do not meet these stringent requirements and can be cracked in a few days time and many could be cracked in minutes! Of those organizations that do have strict enough password policies, many users simply resort to writing their passwords down on sticky notes and slap it on their monitor because they're too difficult to remember. The reality is that we have now reached an era where commodity computing power has exceeded the average human's ability (or willingness) to remember sufficiently complex passwords.

What is needed is an authentication protocol that protects moderately complex passwords that humans can deal with. However, such a strong authentication technology has coexisted with LEAP from the beginning but few users deployed it thinking they were safe in deploying LEAP. These users ignored the fine print that they needed to enforce a strong enough password policy. Complicating matters is that conventional wisdom on what constitutes a strong password is totally outdated when pitted against a modern tool like ASLEAP and it is unfortunate that many enterprise organizations are not aware of this fact.

Theory no more

Although Cisco has maintained for two years that LEAP is secure and still do, LEAP vulnerabilities are no longer just theories. Thanks to Joshua Wright, you don't even need any kind of real hacking skills to exploit them because he has created a weapons grade LEAP cracker called ASLEAP that just about anyone can use. As a result, Cisco has now been forced to suggest that perhaps it would be wise to migrate away from LEAP for organizations that can't or won't enforce strong passwords.

An adversary can now crack the vast majority of enterprise Wireless LANs running LEAP in as little as a few minutes with virtually zero chance of detection because the attack is passive and performed offline using nothing more than commodity computer hardware. To make things even worse, since most LEAP implementations leverage the single sign-on capabilities of most RADIUS servers, the usernames and passwords cracked are usually the same credentials used for Windows Domain Authentication or some other common user database. This means that not only does the hacker gain access to your Wireless LAN, but he also gains possession of most your usernames and passwords that can directly access critical files and applications!

This is even worse than running no encryption at all on your Wireless LAN because a LEAP attack compromises both the network and your user credentials. If you still have any doubts, here is a summary (from the ASLEAP Webpage) of what ASLEAP can do.

- Recover weak LEAP passwords.
- Read live from any wireless interface in RFMON mode.
- Monitor a single channel, or perform channel hopping to look for targets.
- Actively de-authenticate users on LEAP networks, forcing them to re-authenticate. This makes the capture of LEAP passwords very fast.
- De-authenticate users who have not already been seen, doesn't waste time on users who are not running LEAP.
- Read from stored libpcap files, or AiroPeek NX files (1.X or 2.X files).
- Use a dynamic database table and index to make lookups on large files very fast. Reduces the worst-case search time to .0015% as opposed to lookups in a flat file.
- Write *just* the LEAP exchange information to a libpcap file. This could be used to capture LEAP credentials with a device short on disk space (like an iPaq), and then process the LEAP credentials stored in the libpcap file on a system with more storage resources.
- Works on Windows systems with limited functionality due to driver restrictions. Hopefully this restriction will be lifted soon.

Now before you start slamming Joshua Wright, keep in mind that he had kept his promise to hold back on the release of the ASLEAP until Cisco had a chance to develop and release a solution for mitigating the vulnerability of LEAP. It was already possible to break LEAP for years well before the release of ASLEAP but you had to have some hacking skills.

In April 2004, Cisco released EAP-FAST and then Mr. Wright released ASLEAP. Unfortunately, the vast majority of large enterprises have yet to even begin to evaluate this new protocol because it is so difficult for any large organization to make any kind of enterprise wide change. We now have a dire situation where a weaponized form of a LEAP cracker is readily available for anyone to use but few organizations have migrated to a secure form of EAP such as PEAP.

From a hacker's perspective, LEAP is a far more tempting target than even the weak WEP protocol because you can usually crack LEAP in 1/10th the time it does to crack WEP and you get two prizes for cracking LEAP instead of one prize for cracking WEP. Ironically, try cracking WEP on a typical wireless LAN and you might find yourself sitting there for days waiting to collect your hundred million packets. Sure you can theoretically do it in 30 minutes if the traffic is going full throttle, but I doubt that you can do it in less than 10 hours in most cases.

Cracking LEAP with ASLEAP on the other hand can be done in minutes because you can literally kick someone off their wireless connection, watch their re-authentication session, and then perform a password crack on the fly for most user credentials! Worst case, the hacker can spend a few minutes recording some LEAP authentication sessions, go home and dump the output on to his more powerful desktop computer with a massive four gigabyte

database containing pre-computed hashes corresponding to potential passwords, then go to sleep and wake up in the morning with a pile of clear text user credentials on his desktop PC. Most likely, it won't even need all night since ASLEAP is able to compare about 45 million passwords in a single second on meager hardware.

Better alternatives to LEAP

As LEAP began to gain a massive foothold on the enterprise market, a superior form of EAP called EAP-TLS (Transport Layer Security) was readily available and was completely password cracking resistant because it didn't rely on user passwords. EAP-TLS relied on digital certificates on both the Server and the Client end to facilitate mutual authentication and secure key exchange. Unfortunately, the need for a PKI (Public Key Infrastructure) deployment on the server end and the installed user base was too great a barrier for many organizations.

As a result, Funk Software in conjunction with Certicom proposed a new IETF standard called EAP-TTLS (Tunneled Transport Layer Security) to ease the deployment requirements by producing a standard that only required digital certificates on the authentication server end. Digital certificates were no longer needed for the client end which posed the biggest deployment barrier of all.

Similarly Microsoft, Cisco, and RSA collaborated and created their own "lite" version of EAP-TLS called PEAP which in principal was the same as EAP-TTLS and also alleviated the need for client side certificates. As time went on, PEAP became the dominant standard because Windows XP Service Pack 1 bundled a PEAP client with the operating system and Funk made their products compatible with PEAP.

One might wonder why people are not flocking to PEAP authentication which eliminates dictionary attacks against your sensitive user credentials. It may be a combination of ignorance, the tendency to remain with the status quo, or Cisco's marketing power that tells people that LEAP is safe. Many organizations don't want to deploy a digital certificate on their authentication server because of the $300/year price tag of a publicly trusted digital certificate nor do they want to build their own Certificate Authority server or chain of servers for an in-house PKI. Regardless of the reasons, there is still a massive user base using LEAP authentication for their Wireless LAN implementations and it's a huge disaster waiting to happen. Cisco has responded to the threat of LEAP hacking and the reluctance of most of their customers to adopt PKI-based PEAP with their so-called "PKI-free" protocol EAP-FAST which has even more problems in security and ease of deployment.

The bottom line on LEAP

Cisco LEAP authentication is a huge security risk in enterprise wireless LANs. So much attention in wireless LAN security or security in general is given to the encryption component of security that the authentication component is often neglected. If your wireless LAN is running LEAP and this document doesn't scare the living day lights out of you, it should. Move to PEAP, EAP-TLS, or EAP-TTLS.

A primer on Cisco EAP-FAST authentication

Introducing EAP-FAST

With the threat of ASLEAP looming, Cisco created their new Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST) protocol and submitted it to the IETF. EAP-FAST as proclaimed by Cisco marketing would be "as secure as PEAP" and "as easy as LEAP"! EAP-FAST achieves this by creating the same secure encrypted tunnel used to protect user credentials during the authentication session as PEAP without the need for any kind of PKI on the client end or even the server end. Naturally, I immediately became skeptical of that claim. Of course there are other ways such as secret key cryptography of achieving a secure key exchange without PKI, but none are ever as elegant or as easy to deploy in large scale. Is EAP-FAST the exception?

EAP-FAST phases

EAP-FAST under normal operation is just like PEAP and has two phases. Phase 1 sets up a secure encrypted tunnel and Phase 2 is a MS-CHAPv2 session that verifies the client to the authentication server. Since MS-CHAPv2 is widely known to be very weak against dictionary attacks, the encrypted tunnels established in Phase 1 provides a safe environment for the MS-CHAPv2 session. The difference is that EAP-FAST uses a PAC (Protected Access Credentials) shared secret to set up the tunnel where as PEAP uses the server side digital certificate to set up a TLS tunnel (similar to how secure Web servers work). A unique user specific PAC file is generated from a single EAP-FAST Master Key on the authentication server for each and every user. Distributing the PAC can be achieved by an optional "Phase 0" (AKA automatic provisioning) or by other out-of-band methods such as sneaker net, protected admin-only file share, or user specific restricted directory.

Marketing vs. reality

After carefully reading through the IETF draft of the EAP-FAST protocol submitted by Cisco, it became fairly obvious that Cisco's marketing claims were not entirely accurate. Although it was technically true that EAP-FAST "could" be as secure as PEAP and it "could" be as easy as LEAP, Cisco's marketing left out the fine print and omitted the fact that you just couldn't have both at the same time.

The fact of the matter is, in order for EAP-FAST to truly be as secure as PEAP, it would have to run in "server-side authentication Diffie-Hellman mode" in "Phase 0" which ironically requires a server side Digital Certificate. As you recall, the very need for a server side Digital Certificate was the very thing that scared people away from PEAP in the first place. Cisco says that you could resort to other "out-of-band" methods of PAC provisioning which doesn't require a server side certificate but the practicality of manual PAC provisioning is rather dubious. The EAP-FAST PAC refresh mechanism on the other hand does automatically provision keys in a secure manner but it can only be used to maintain the PAC but does not address the need for initial PAC provisioning.

In order to be as easy as LEAP, EAP-FAST had to run "Phase 0" in anonymous Diffie-Hellman mode. An anonymous Diffie-Hellman key exchange by definition means that you can't verify who is on the other end. In this case, a hacker can pose as your access-point and authentication server in Phase 0 and wait for an unsuspecting user to connect at phase 0 and wait for the clear text username and the hashed passwords. Since the server is obviously a fake, it would fail the server challenge and phase 0 would be aborted by the client. However, enough of the MS-CHAPv2 session has been captured to perform an offline ASLEAP attack. If the user's password is not extremely complex which is usually the case, then game over and the hacker gains both user credentials and access to your wireless LAN.

According to the EAP-FAST IETF draft paper, if such a man in the middle exploit is attempted, the user is to immediately change their password. However, there is no indication that the EAP-FAST client will automatically warn the user and administrator or force the password change. At least running EAP-FAST with this convenient but weak form of automatic PAC provisioning is still significantly more secure than running LEAP for two reasons.

- First, PAC provisioning is only done once to set up the PAC secret between the server and client and all subsequent EAP-FAST sessions skip "Phase 0". LEAP on the other hand is vulnerable each and every time a user authenticates with the radius server during the wireless LAN authentication.
- Second, even during a phase 0 anonymous DH session, the attack must be active which exposes the hacker to detection and is significantly more risky. A LEAP attacker on the other hand can perform the exploit with virtual impunity.

Even though this is a significant improvement over LEAP, EAP-FAST can never be as secure as EAP-TLS, EAP-TTLS, or PEAP while running in this mode. EAP-FAST does provide a faster authentication session because it uses symmetric cryptography instead of the asymmetric cryptography that EAP-TLS, EAP-TTLS, or PEAP uses but that's hardly an advantage. I have even tried PEAP authentication on 266 MHz PDA devices which is the slowest platform you would run EAP on and I can't see any significant delays with PEAP. I really doubt you will ever miss a few milliseconds on your notebook when authenticating with PEAP for wireless LAN access. Speed in deployment is what I'm concerned about and this is where EAP-FAST falls on its face.

EAP-FAST deployment troubles

The deployment of EAP-FAST is marketed "as easy as LEAP" but the reality is not so simple. According to Cisco's own EAP-FAST deployment guide, you cannot completely rely on automatic provisioning of PAC files because it is susceptible to an active attack. The following is an excerpt from that deployment guide.

Note: *Because transmission of PACs in phase zero is secured by MS-CHAPv2 authentication and MS-CHAPv2 is vulnerable to dictionary attacks, we recommend that you limit use of automatic provisioning to initial deployment of EAP-FAST. After a large EAP-FAST deployment, PAC provisioning should be performed manually to ensure the highest security for PACs. For more information about manual PAC provisioning, see Manual PAC Provisioning.*

Source: Cisco EAP-FAST deployment guide

As you can see in the fine print, EAP-FAST deployment is not that straight forward. You must eventually resort to manual provisioning of user specific PAC secrets that must be deployed with the greatest secrecy and care. The secret must even be kept from other legitimate users or else the PAC is still compromised because you wouldn't want multiple users knowing the same PAC.

After reading the Manual PAC Provisioning section from the same Cisco document, it became rather amusing to see the amount of work it takes to deploy EAP-FAST in a secure manner. Having plenty of experience in deploying EAP-TLS or PEAP, I can tell you right now that EAP-FAST manual PAC provisioning is the most labor intensive way of deploying secure authentication. In the case of PEAP, trust of the PEAP server's digital certificate is inherent if the certificate was purchased from a public Certificate Authority.

Even if you didn't want to spend the $300 a year to buy a digital certificate for your PEAP server, you could always build your own Certificate Authority or self sign a digital certificate which could be deployed automatically using Active Directory. If your company runs Windows NT domains or some other non-Microsoft user directory which doesn't support automatic deployment of a Root Certificate, then the public certificate ".cer file" of the PEAP server could simply be posted on a public intranet page for users to manually install in their root CTL (Certificate Trust List). There is absolutely no fear of the public certificate falling in to the wrong hands because it only contains the 1024-bit public key portion of the certificate which is practically impossible to crack.

With EAP-FAST user specific PAC files, you literally have to generate and manage thousands (one for every user) of unique user-specific private keys where you can't even afford a single compromise of a single key. You can forget about posting that on a public intranet server let alone automatically deploying it with Active Directory group policies. Each PAC is unique and must be manually imported in to the Cisco ACU (Aironet Client Utility) on each end user's laptop. You can see why I chuckle when Cisco claims EAP-FAST is "easy as LEAP" because it isn't even as easy as EAP-TLS in a managed environment.

More limitations on Cisco EAP-FAST

Cisco EAP-FAST requires the use of newer Cisco Wi-Fi adapters since the older cards don't support it. Cisco EAP-FAST also requires the use of a very expensive authentication platform using Cisco ACS (Access Control Server) which ironically isn't as flexible or easy to manage as Microsoft's built in RADIUS server IAS (Internet Authentication Service) and this is speaking from a lot of experience dealing with both platforms. The Cisco client also doesn't support "machine login" which is a way for a computer to log on to the network **before** the user signs on to Windows. For Enterprise deployments this is extremely important because of the need for logon scripts and group policies to function properly. Cisco's own Website instructs users to use the Microsoft Wireless Client if they wish to implement machine logins.

The bottom line on EAP-FAST

So is Cisco's EAP-FAST an exception to the rule where you can't get away with no PKI to facilitate secure key exchange on a large scale? I think we can safely conclude no, it cannot. After reading through the deployment section, one really begins to wonder if EAP-FAST is really worth all that trouble just to avoid deploying a digital certificate on the authentication server because you don't want to build a PKI Certificate Authority or because you don't want to purchase a $300 digital certificate every year. If you were hoping that EAP-FAST was going to be your savior, it won't happen since it can never be "as easy as LEAP" if it wants to be secure like PEAP. The best solution is to use PEAP authentication.

Self-signed certificates for your RADIUS server

Introduction to self-signed digital certificates

Self-signed digital certificates is a way avoiding the use of public or private Certificate Authorities. They have long been used by developers for the purpose of testing secure Web servers and code signing but have not been used in production systems. Few people know of this method or use it for RADIUS PEAP authentication and it has been difficult to find any documentation anywhere on the Internet or books explaining how to do this.

The concept of self-signed digital certificate is similar to Pretty Good Privacy (PGP) because it doesn't use the Certificate Authority model. Although both PKI and PGP are part of the broader umbrella of PKC, digital certificates were designed to conform to the PKI trust model made up of centrally trusted CAs while PGP used a freeform peer-to-peer method of establishing trust.

For example, a PGP user would generate their own public and private key pair and then post the public key to their own public Website for all to verify. Because of this model of establishing trust, there is no need for a public or private CA which is the biggest impediment to secure authentication protocols such as SSL and PEAP.

To create a self-signed digital certificate, one would simply use a utility (shown in next section) to generate a digital certificate with a digital signature. The difference here is that instead of using an external trusted CA (analogous to a Notary) to sign the digital certificate, the utility would simply sign the certificate itself.

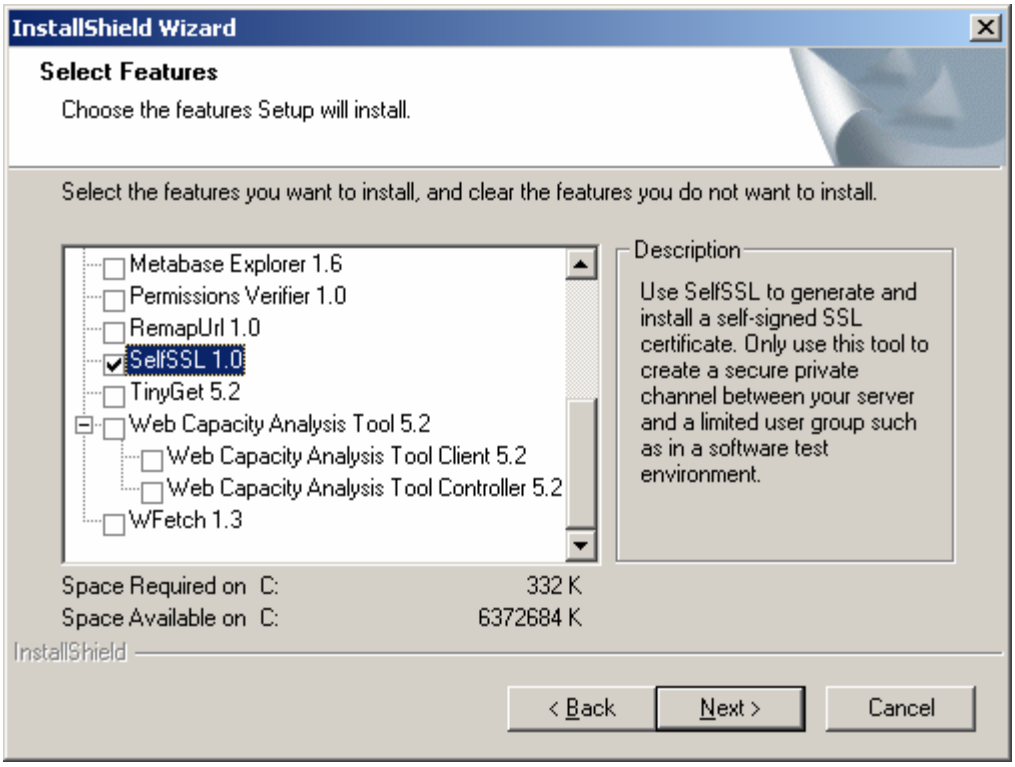
Once the digital certificate is generated, a pubic version of the digital certificate containing the only the public key called a "root certificate" can be exported and be made publicly accessible. The root certificate can be distributed by any means (even on a public Website) without fear of compromising the certificate since the private key is kept private. As with any PKC technology such as PGP or PKI, there is no practical method of deriving the private key from the public key. Once a self-signed digital certificate, users can securely authenticate against that RADIUS server using PEAP authentication.

Microsoft IIS 6.0 Resource Kit

As soon as I thought of using self-signed digital certificates for PEAP authentication, I began looking for a simple utility for creating self-signed digital certificates. After an extensive search, I found within the Microsoft IIS 6.0 Resource Kit an interesting command line utility called SelfSSL.exe which can create self-signed digital certificates. Although it's intended to be used for Microsoft IIS 6.0 SSL Web server testing, it works for many other applications as well including PEAP since the certificate it generates is a standard X.509 certificate. After a quick test in the lab, it became obvious that this was a good alternative to building a PKI Certificate Authority to simplify PEAP authentication. Download a copy of the Microsoft IIS 6.0 Resource kit here

When you install it, you only need to install the 332 KB SelfSSL 1.0 component of the Resource Kit. (**Figure A**)

Figure A



SelffSSL 1.0 Installation Wizard

The SelfSSL.exe tool should work with most RADIUS/AAA Authentication Servers and I've verified this on Microsoft IAS server. On your Authentication Server, open up a command prompt and go to the directory where you installed it (default -- **C:\Program Files\IIS Resources\SelfSSL)**. You then type the following command.

```
selfssl /N:CN=ServerName.YourDomain.com /K:1024 /V:1825 /S:1 /P:443
```

- **/N:CN** should be set to your ServerName and your fully qualified domain name.
- **/K:** typically set to 1024. 1024 is the number of bits allocated to the RSA key.
- **/V:** is the number of days before the certificate expires. 1825 days is 5 years.
- **/S:** is the site number in IIS.
- **/P:** is the TCP port number. 443 is the standard SSL port.

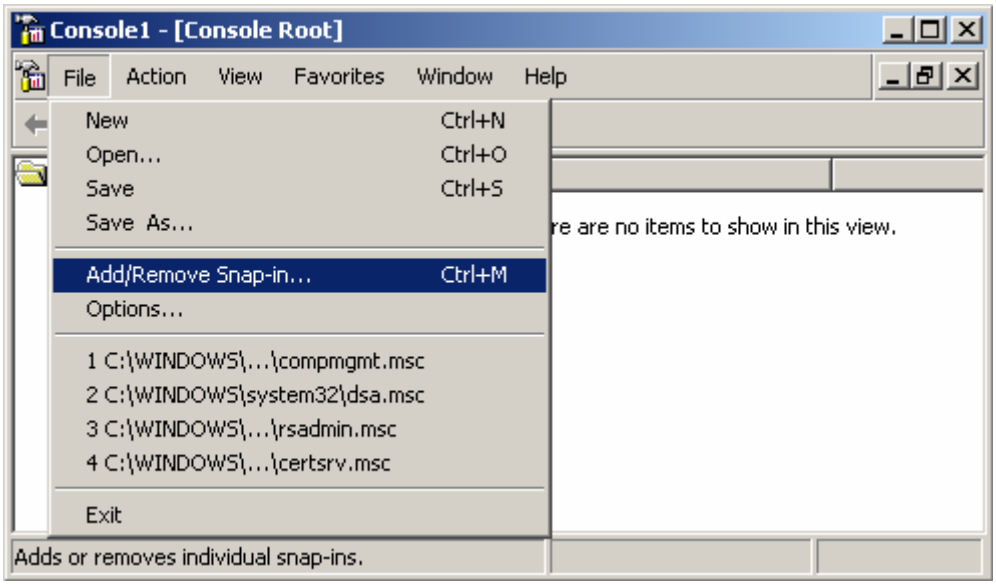
Note that **/S:** and **/P:** are irrelevant in our case since you don't need IIS running on your Authentication Server. As a general rule of thumb for security sake, you run as few services on your server as possible. If you don't have IIS installed, executing the SelfSSL command as shown above will end with an error message "Error opening metabase: 0x80040154". That just means the IIS site was not found but **you can ignore that error message** since the Certificate you need for PEAP authentication will have already been generated.

Creating the root certificate

Once the digital certificate has been generated on your authentication server, you will need to export the root certificate for this Self Signed Certificate. The digital certificate is different from the root certificate. The digital certificate contains the public and private key pairs. The root certificate only contains the public key and a self proclamation that "I am a root certificate". You will need this root certificate for publication on a Web-server or file-server for manual root certificate deployment or you can import it in to your Active Directory Group Policy for automatic root certificate Deployment.

To begin, you'll need to open an MMC console by clicking Start | Run. Then type "mmc" and OK. You will see the following console appear (**Figure B**). From there, you'll click "ADD/Remove Snap-in...".

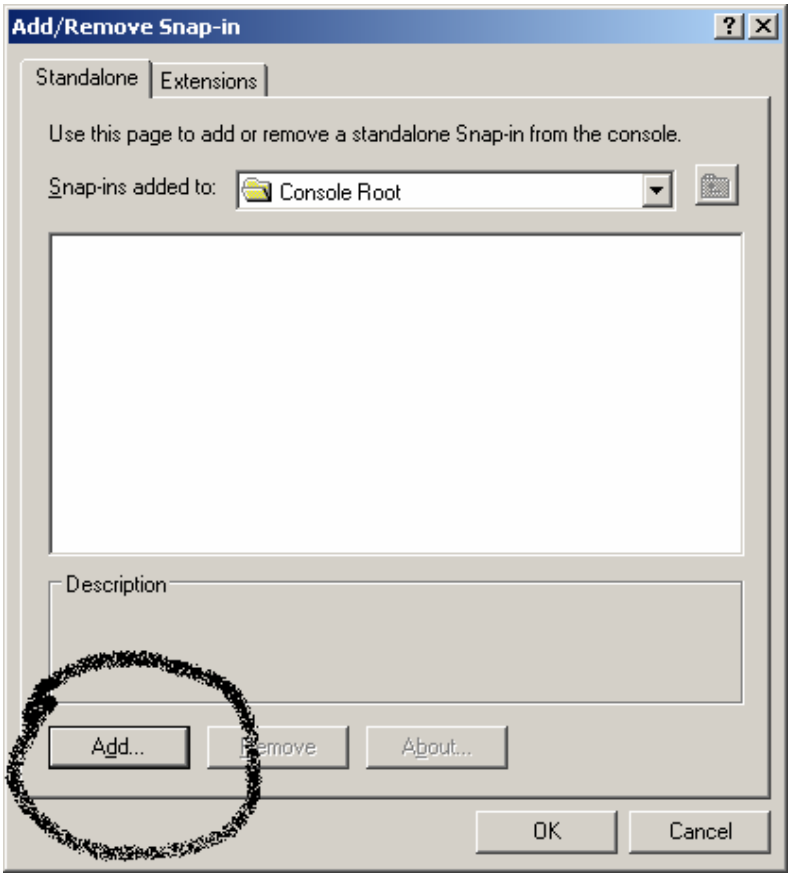
Figure B



MMC Console

You'll then see this screen (**Figure C**). Click on the "ADD" button.

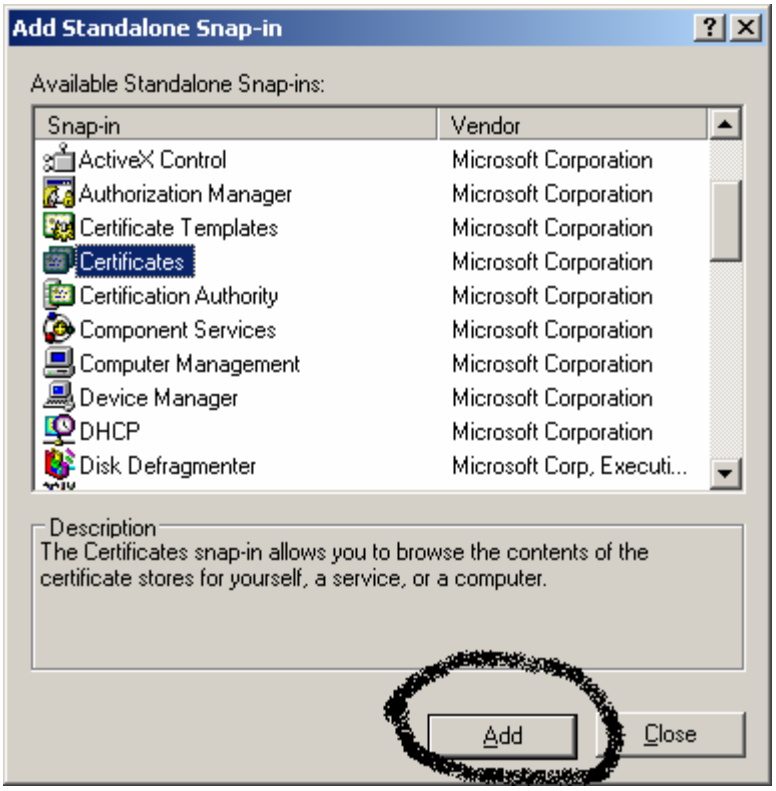
Figure C



Add/Remove Snap-in

On this screen (**Figure D**), highlight "Certificates" and click on "Add" again.

Figure D



Certificates

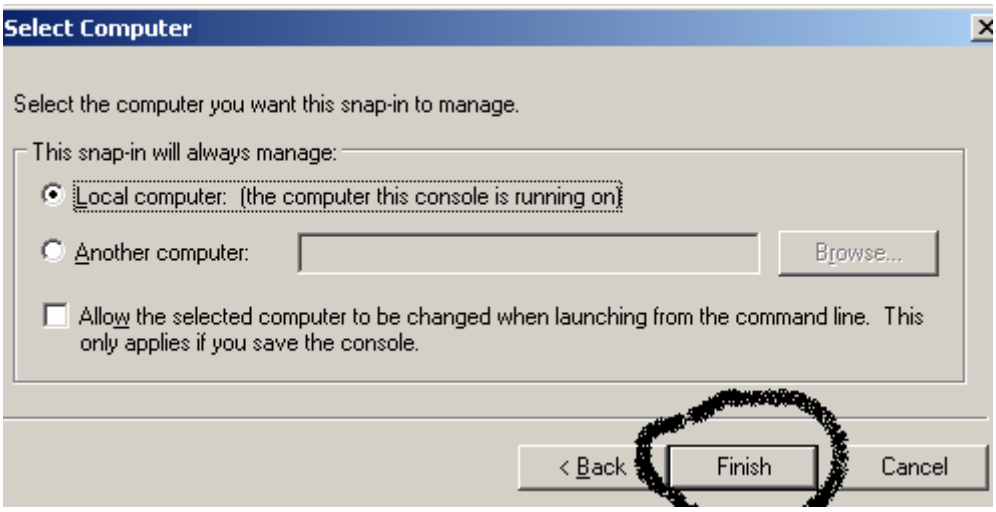
Select "Computer account" and click "Next". (**Figure E**)

Figure E

Computer account

Then select "Local computer" as shown below in **Figure F** and click "Finish".

Figure F



Local computer

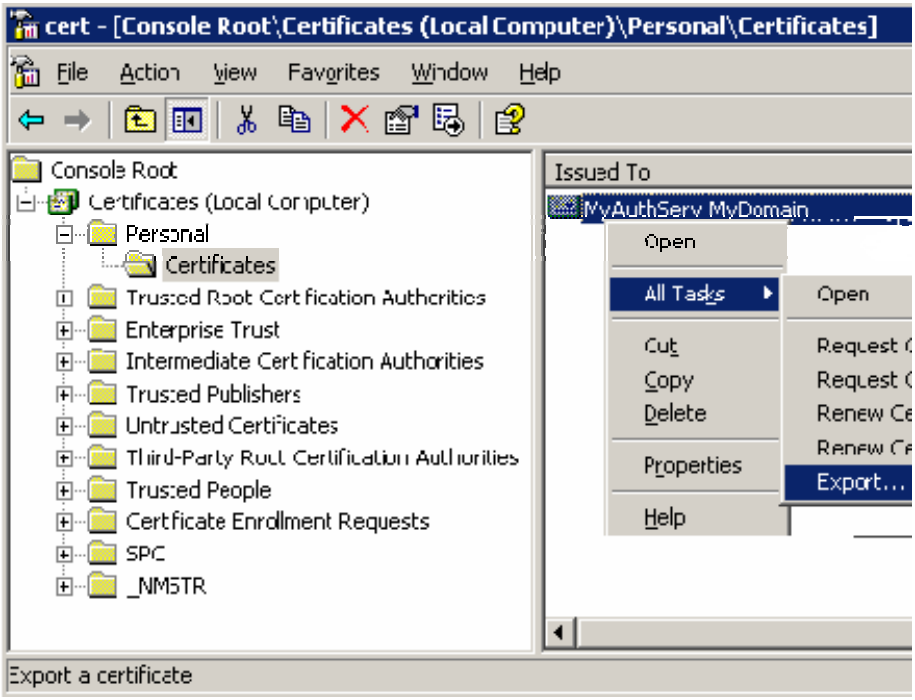
You will see the resulting console appear. (**Figure G**)

Figure G

Console root

Expand "Certificates (Local Computer) to reveal the following. Right click on "MyAuthServ.MyDomain" or whatever you used for your SelfSSL "/N:CN" argument, hit "All Tasks" and then choose "Export". (**Figure H**)

Figure H



Export

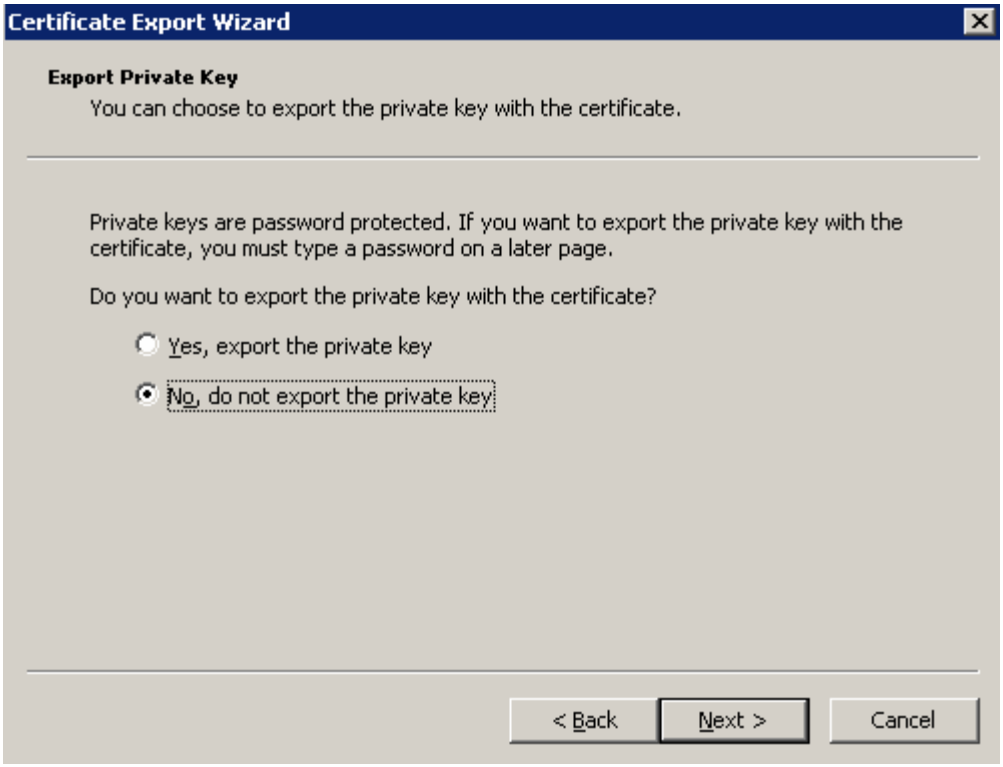
You will see the following wizard (**Figure I**). Choose "Next".

Figure I

Certificate Export Wizard

For this step, make sure you DO NOT export the "Private Key" because that must be kept private on the server. If you use the "Yes, export the private key" feature, that allows you to make a backup of the digital certificate but you want to guard that file in a protected area. Anyone who gets that file compromises your digital certificate because they now have a copy of your private key. Exporting the private key also lets you take that digital certificate and copy it to a redundant RADIUS server so you can import it there without having to generate a second key. If you have more than one RADIUS authentication server, make sure you copy the certificate over and don't generate a second key unless you want to complicate deployment matters by having to deploy two root certificates. (**Figure J**)

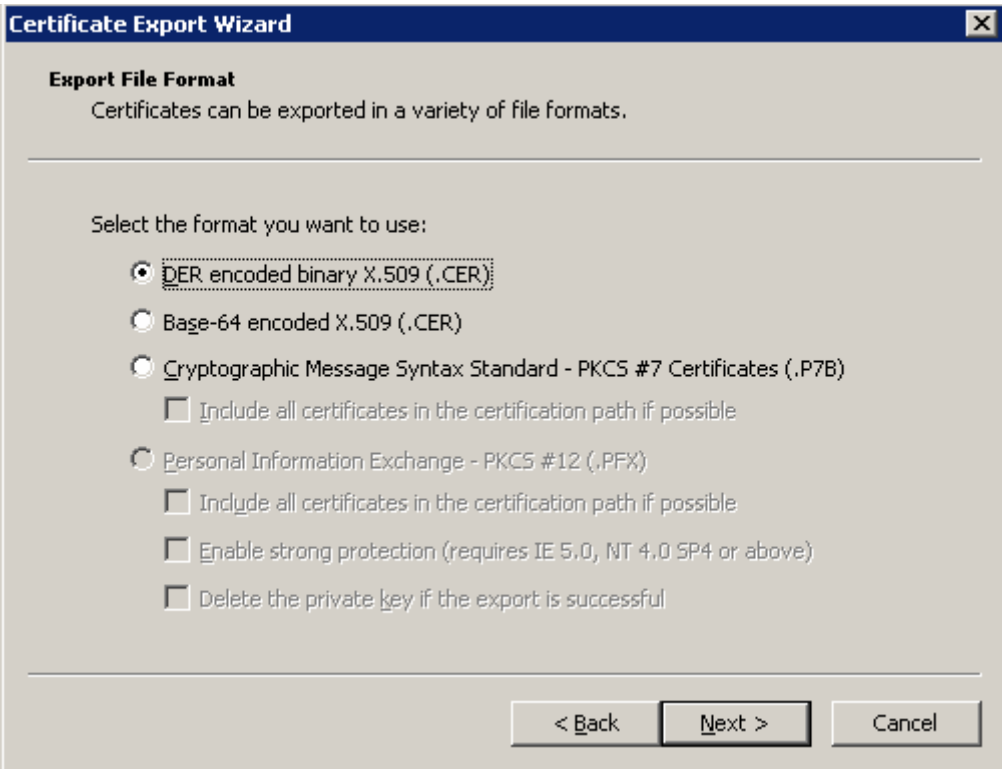
Figure J



Not the private key

Use the "DER" format because it is compatible with Windows and Windows Mobile devices (**Figure K**). Windows doesn't care what format it's in but Windows Mobile does.

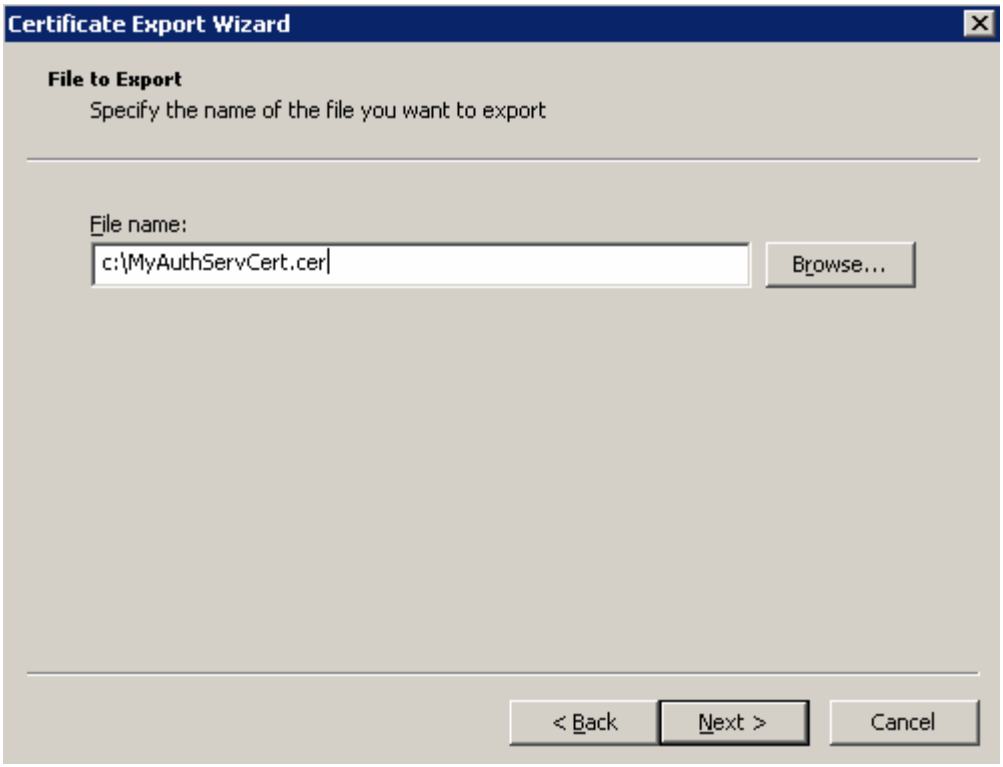
Figure K



File format

Give the certificate a path and file name. (**Figure L**) You'll need to note the name for later use.

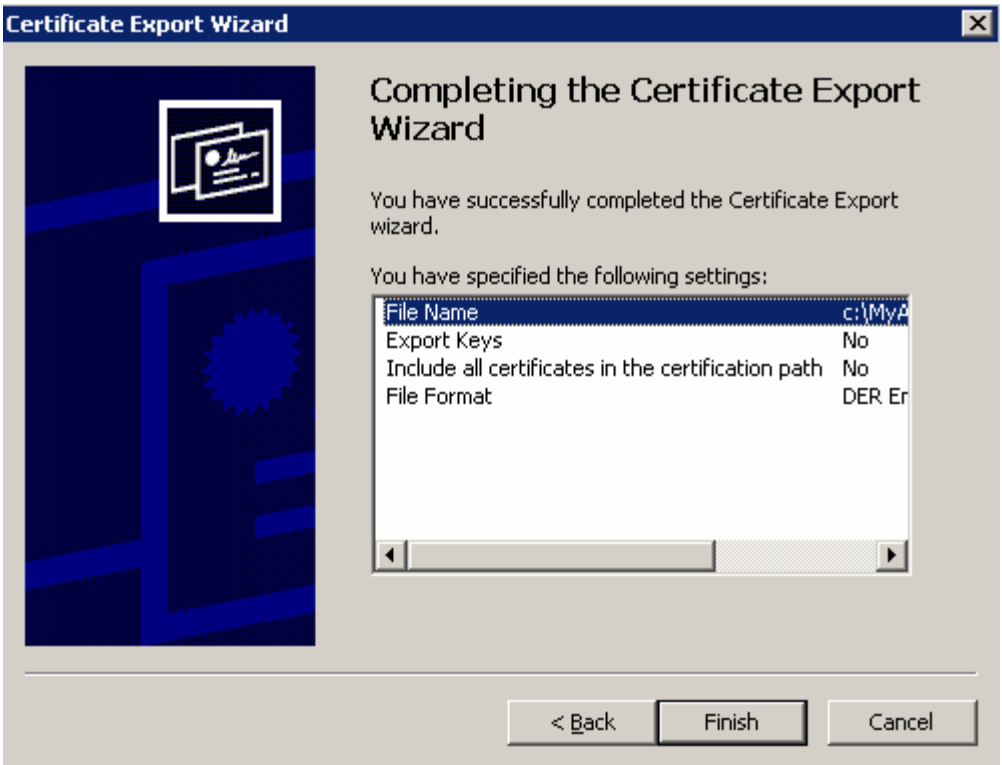
Figure L



Path and file name

Hit "Finish" and you've just exported your Self Signed root certificate to a file. (**Figure M**)

Figure M



Finish

Now you're have a self-signed root certificate ready to be deployed to the clients automatically or manually along with the digital certificate on your authentication server ready to use. We'll discuss how you actually use this certificate on our Microsoft IAS RADIUS server configuration guide.

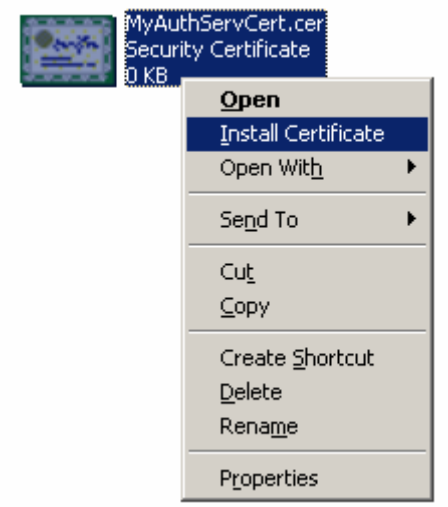
Manual Root Certificate deployment for Windows

Only use this section if you don't have Microsoft Active Directory to automatically deploy your "root certificate" to your user's Certificate Trust Lists (CTL). This article assumes that you have set up some way of distributing your "Root Certificate" either by posting it on an internal Intranet server, a public Internet server, or internal file server. You don't need to worry about this certificate falling in to the wrong hands so long as you didn't include the private key when you exported the certificate, but you might still want to keep the distribution of your root certificate internal.

Start by copying the Certificate Authority Certificate to your Laptop, Desktop, or PDA and use the following procedure.

Right click on the file "MyAuthServCert.cer" and click "Install Certificate". (**Figure N**)

Figure N



MyAuthServCert

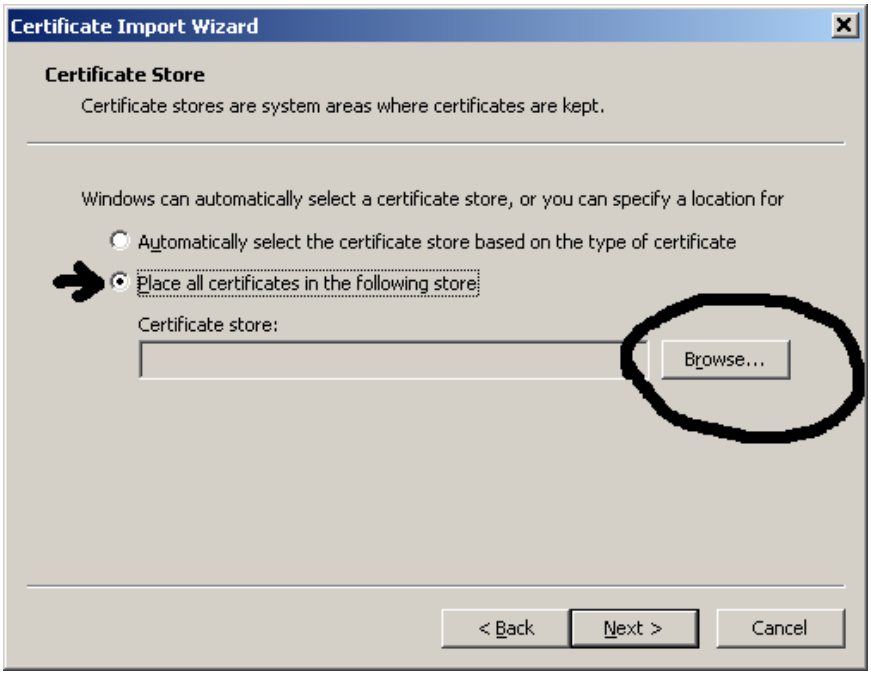
Click "Next" on the welcome screen shown in **Figure O**.

Figure O

Certificate Import Wizard

Choose the second option and click "Browse." (**Figure P**)

Figure P



Certificate Store

Click on "Show physical stores" and expand "Trusted Rood Certification Authorities" and select "Local Computer". **Make sure** you follow this particular instruction very carefully to put the cert in the right place! (See **Figure Q**)

Figure Q

Select Local Computer

Click OK, Next, and then Finish to complete this phase.

Note that this same "Root Certificate" works on Pocket PC Windows Mobile 2003 (or above) and Windows CE 4.2 as well. You simply need to download the "root certificate" and double tap on the file. It will prompt you to install it and all you need to do is click "Yes" or "Ok". This technique does not work on PALM based devices because they don't support 802.1x and PEAP authentication.

It is also possible to get modern versions of Mac OS X or Linux (with the proper supplicant software) working as well and it works in the same manner. If you're running the Cisco Aironet Configuration Utility (ACU) client on Windows, both the automatic and this manual method of installing a root certificate works though the Cisco Wireless Client can't be auto configured through group policy and it doesn't support machine logon.

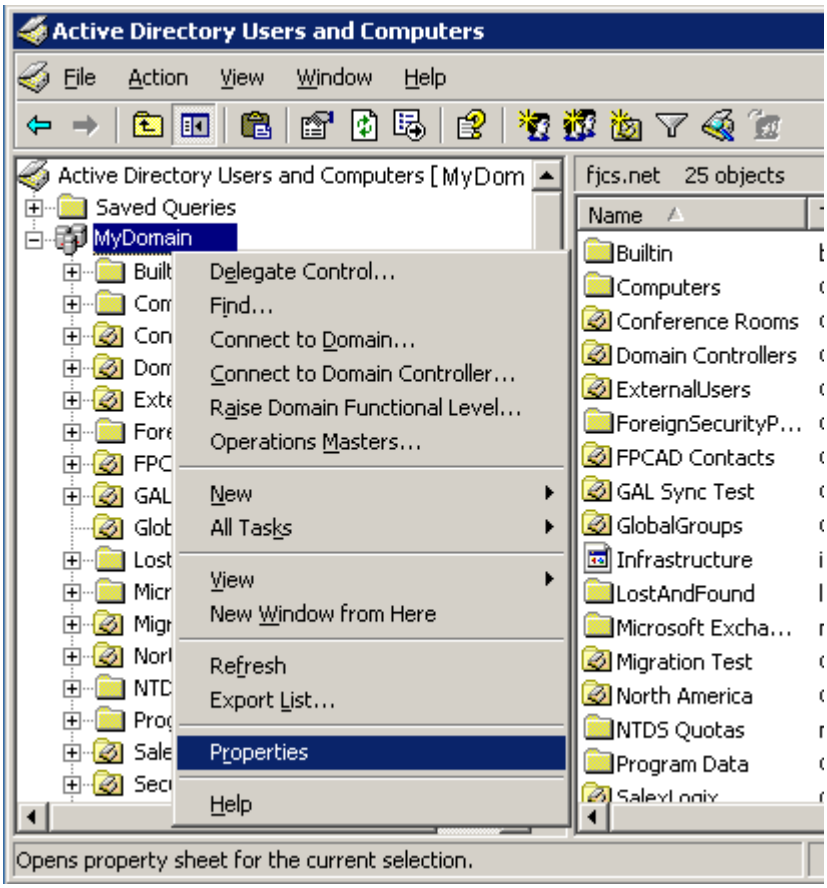
Automatic Root Certificate deployment

Before any Digital Certificate can be used, the signing authority (self-signed in this case) of that certificate must be in the Certificate Trust List (CTL) of the client computer. The technique described in this section works for both Cisco based clients and the native Windows Wireless Zero Configuration (WZC) clients that's built in to Windows XP SP1. If your organization is running Windows AD (Active Directory), there is an extremely simple way of globally inserting a Root Certificate in to the CTL of all users within the AD. If your organization doesn't run Active Directory, skip straight to the Manual Root Certificate deployment.

To get started, you need to fire up your group policy editor by opening up "Active Directory Users and Groups" as a Domain Administrator. You would normally want the Root Certificate to be deployed to the entire domain, but you can also limit the deployment to a certain Organizational Unit that contained a certain class of users. In the rest of my examples in this document, we'll assume that you are deploying the Root Certificate and Wireless PEAP Configuration to your entire Active Directory.

Right click on your domain and click "Properties" as shown in **Figure R**.

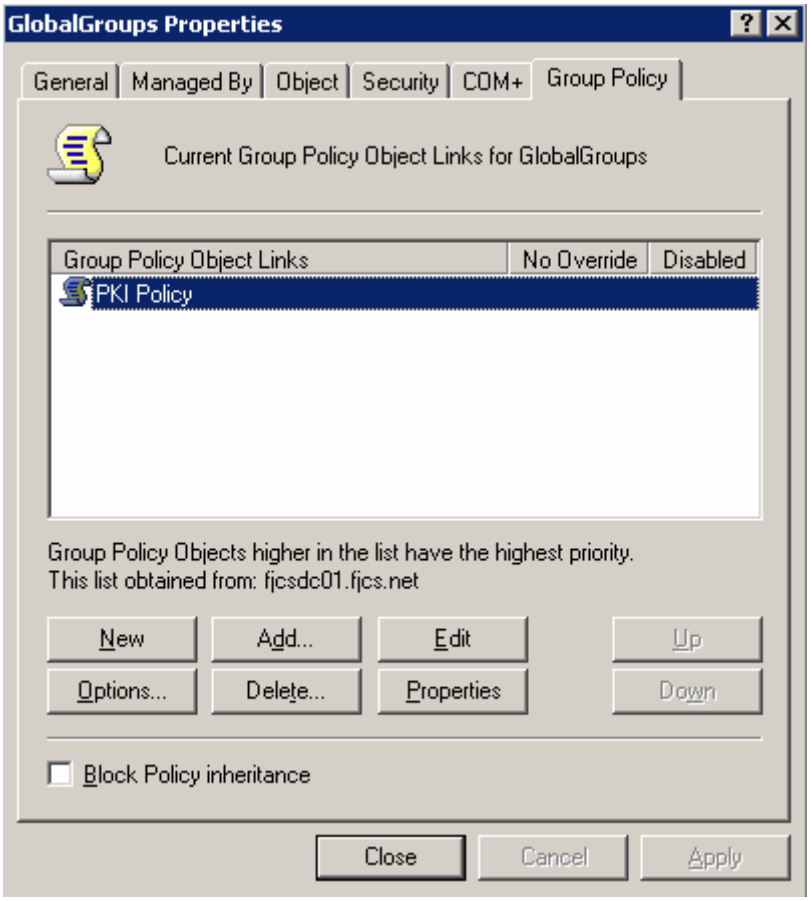
Figure R



Properties

Click on the "Group Policy" Tab. Click "New" and make a new policy called "PKI Policy", then click "Edit". (**Figure S**)

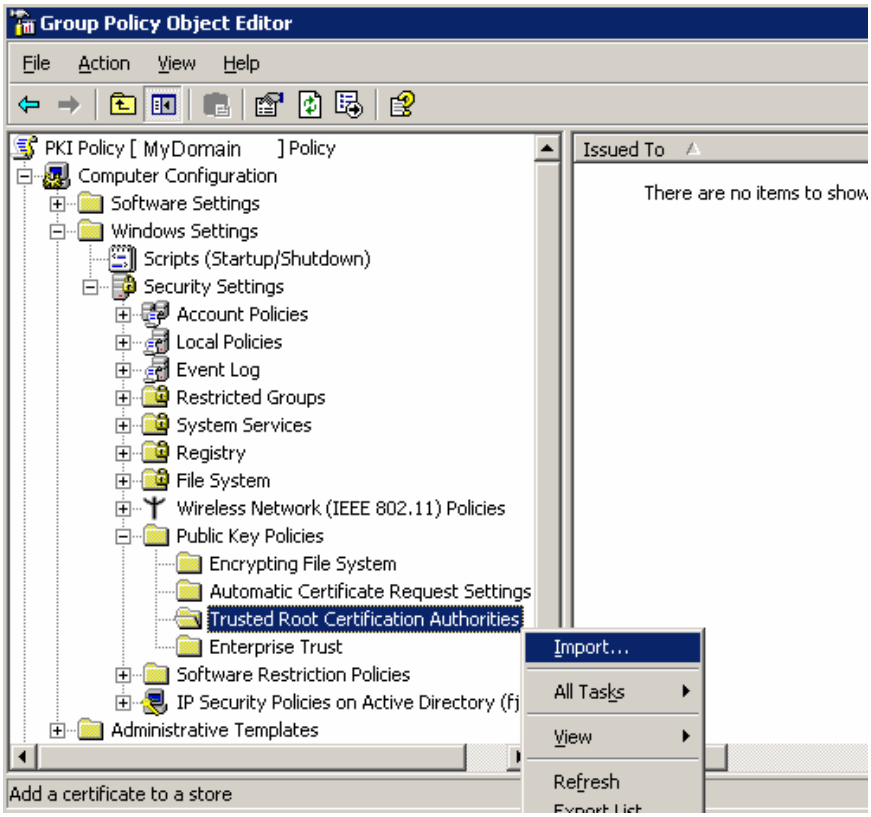
Figure S



PKI Policy

Expand "Computer Configuration" as shown in **Figure T**. Then right click on "Trusted Root ..." and click "Import".

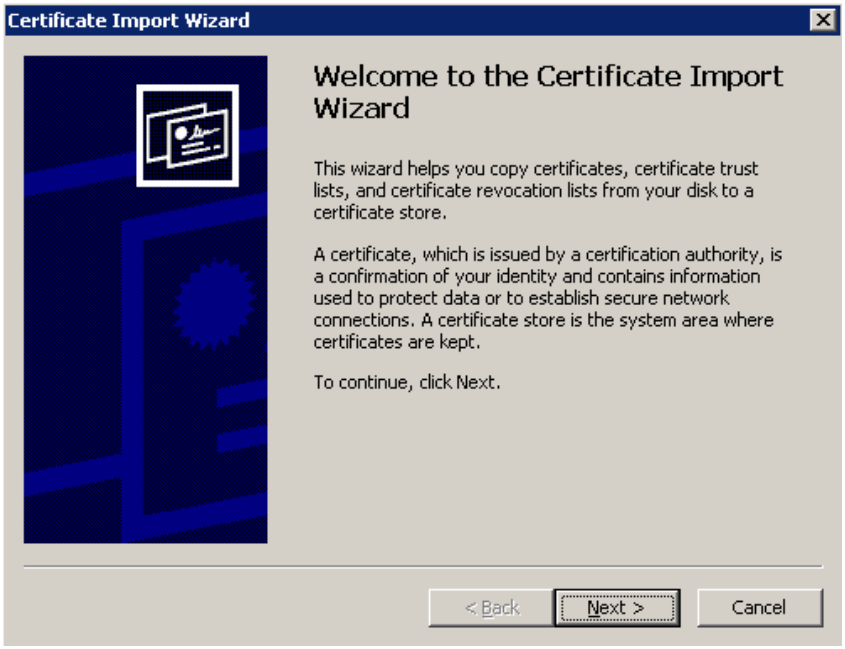
Figure T



Group Policy Objective Editor

Import the Self Signed Root Certificate (**Figure U**)and continue with "Next".

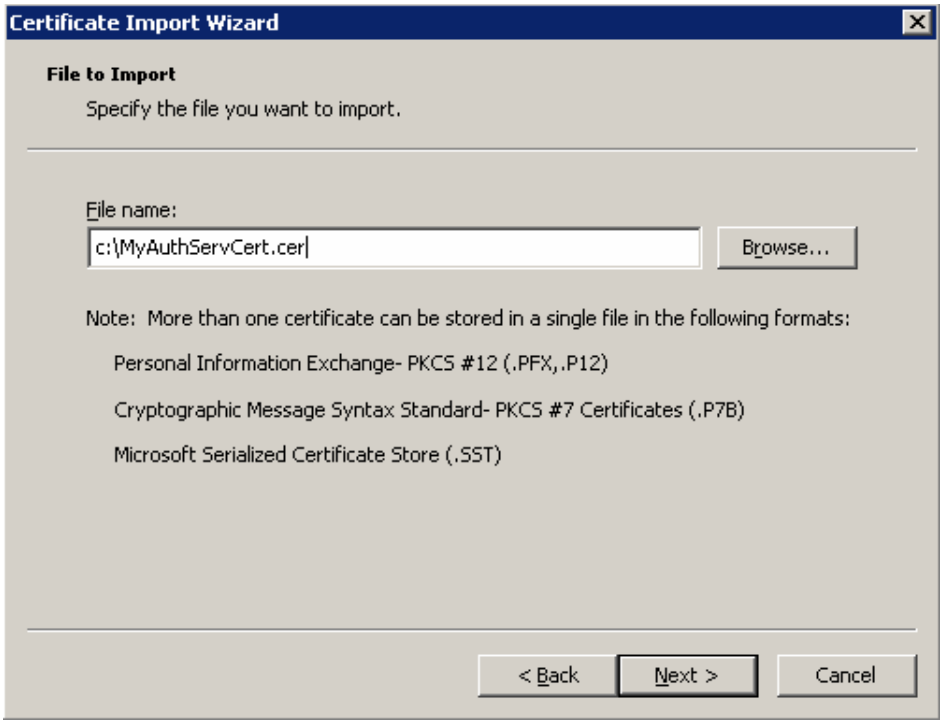
Figure U



Certificate Import Wizard

Assuming you've copied your "Root Certificate" to the C:\ directory of the machine you're editing the group policy on, type in the path and name and click "Next". (**Figure V**)

Figure V



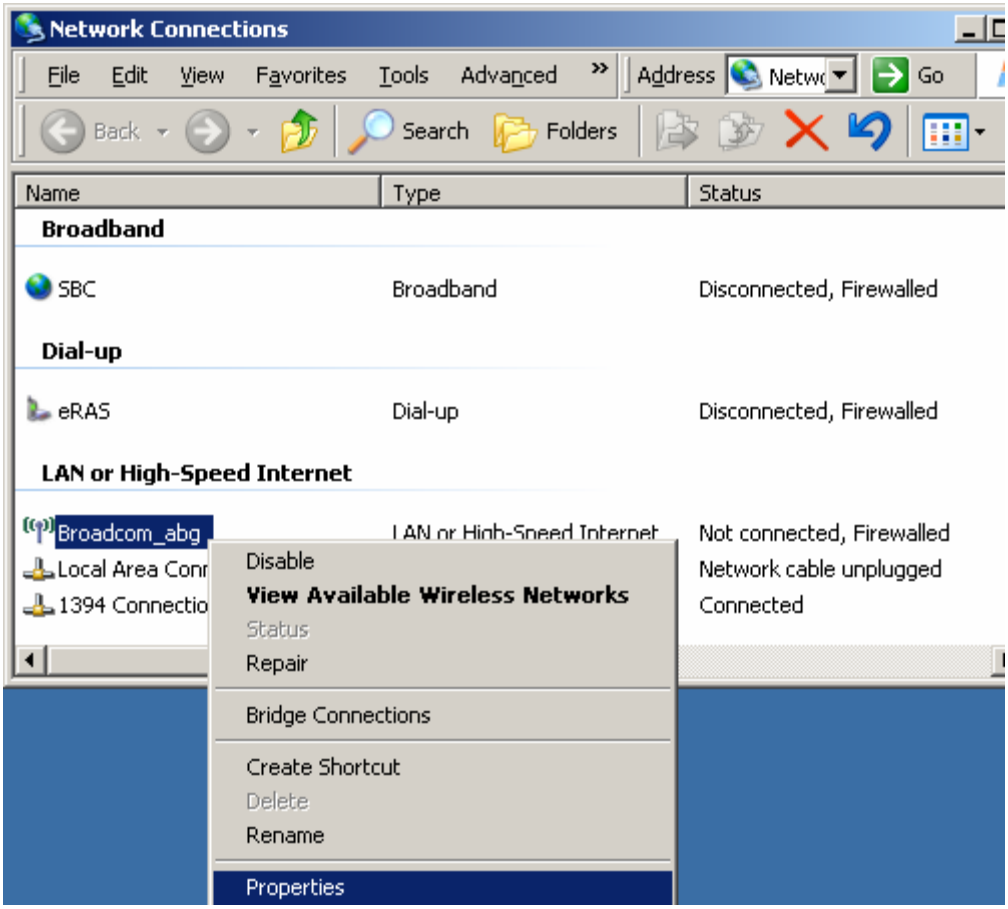
File to import

Click "Finish" on the next screen and close out all of the remaining windows. Once this is complete, your entire Active Directory will "trust" your new "Self Signed" certificate that you self-signed with the "SelfSSL" tool. This exact same technique also works for deploying the root certificate of any PKI Certificate Authority server you build.

Manual PEAP deployment for Windows Wireless Client

Windows Wireless Client formerly known as the Wireless Zero Configuration (WZC) service is Microsoft's built in supplicant (Client) for Wireless Networking. This is the very same client that can be configured automatically by Active Directory Group Policy if you're running a minimum of Windows Server 2003 with SP1 or Windows Server 2003 R2. This section will be mostly demonstrated with Windows XP Service Pack 1 and 2. (**Figure W**) You should absolutely forget about Wireless networking before Service Pack 1 for security reasons and besides it doesn't support PEAP anyway.

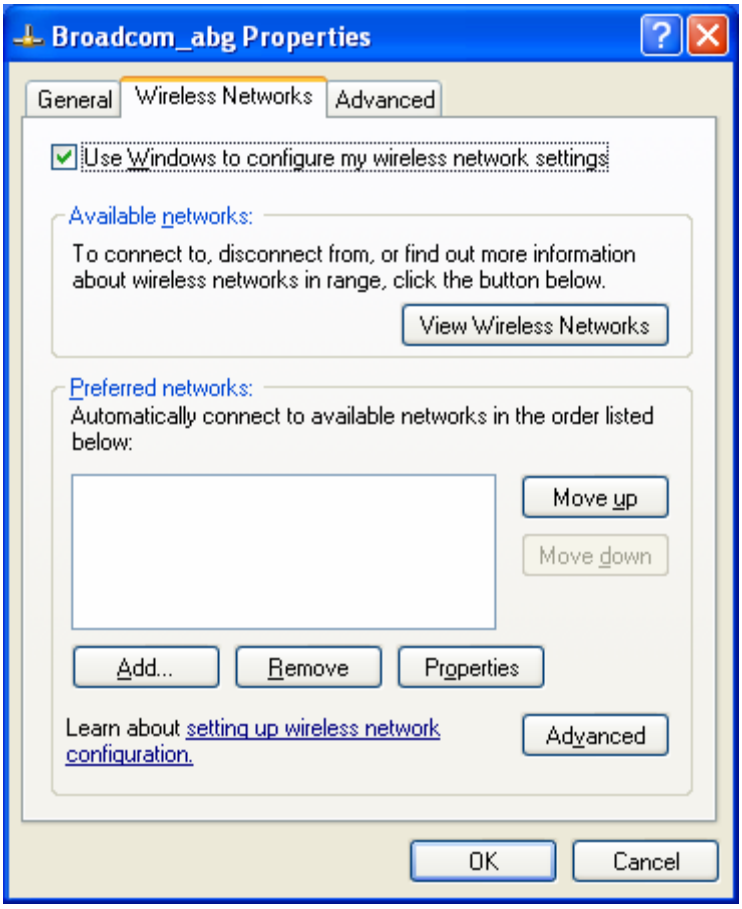
Figure W



Network connections

In either SP1 or SP2, you can configure the Wireless Client by the right clicking on your Wireless Ethernet device under the "Network Connections" folder and then selecting properties. It will take you to the screen shown in **Figure X**.

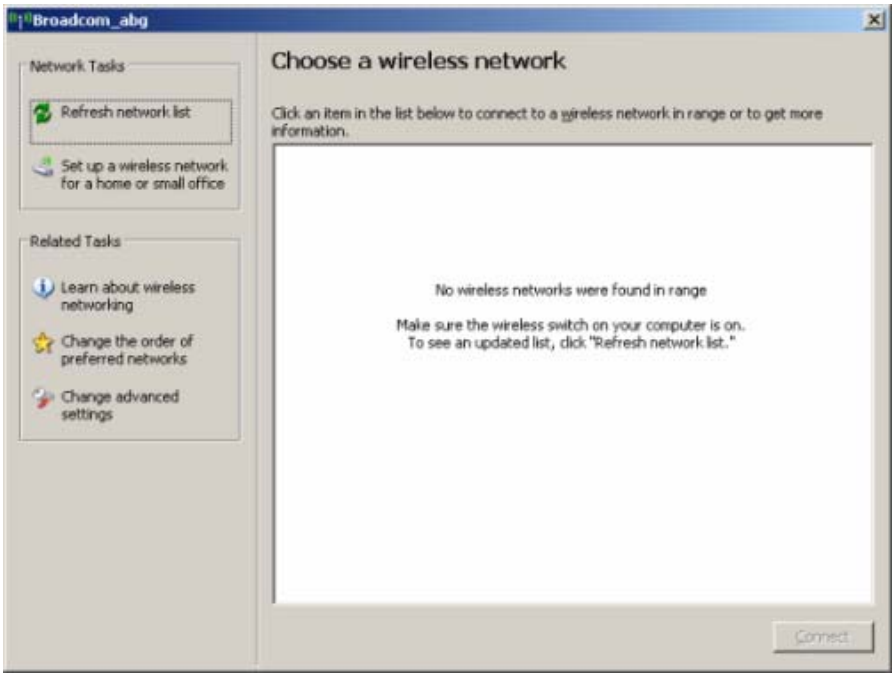
Figure X



Connect to a wireless network

Service Pack 2 has the updated interface shown in **Figure Y** which you can get to by right clicking the Wireless Icon in your system tray on the bottom right of your screen and selecting "View Available Networks". Clicking on "Change advanced settings" will also take you to the screenshot shown in **Figure X**.

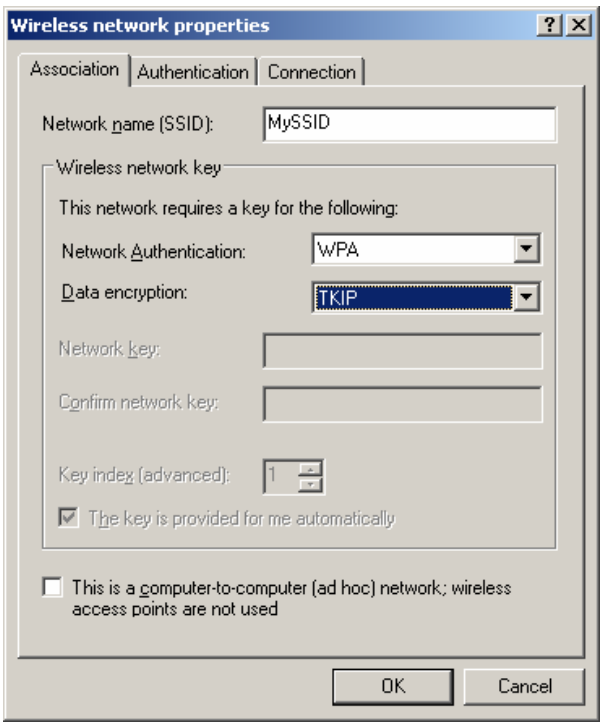
Figure Y



Choose a wireless network

Under XP SP2, you will need to create a profile from scratch by clicking on the "Add" button under the "Preferred networks" section. (**Figure Z**) After you click on "Add", you will see the screen to the left. You simply need to **type in the SSID** that you are using on your Access Points.

Figure Z



Add a wireless connection

For "Network Authentication", select "WPA"

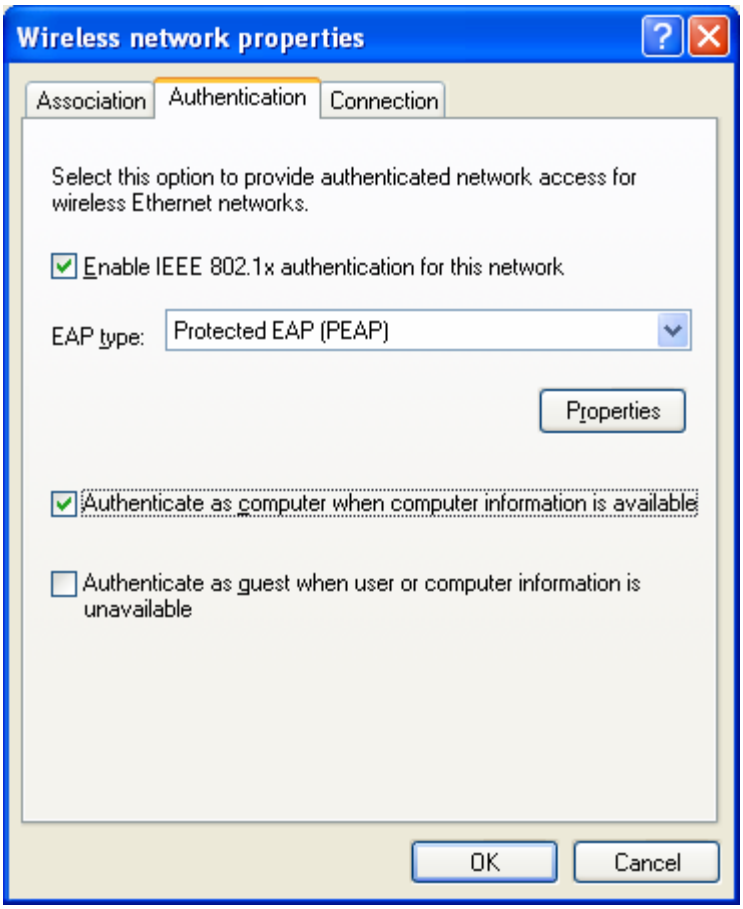
WPA mode requires XP SP2 or Vista. WPA2 is also an option if you installed the additional WPA2 patch for Windows XP or if you're running Vista. Your Access Point also needs to be new enough or have a firmware that supports WPA mode and your Wi-Fi client adapter drivers supports WPA.

For "Data encryption", select "TKIP"

With WPA or WPA2 mode, you have to choose either TKIP or AES. AES is better but not all hardware on the access point or client side can support it so check your hardware capability. AES is clearly the preferred choice if you can get it to work but TKIP is now a minimum security requirement. WEP is no longer valid under ANY scenario even if you're using it with RADIUS and key rotation. This is because the attack methodology for WEP advanced dramatically in 2005.

Switch over to the "Authentication" tab as shown in **Figure AA**. Check the "Enable IEEE 802.1x authentication for this network" and select "Protected EAP (PEAP)".

Figure AA



Wireless Authentication properites

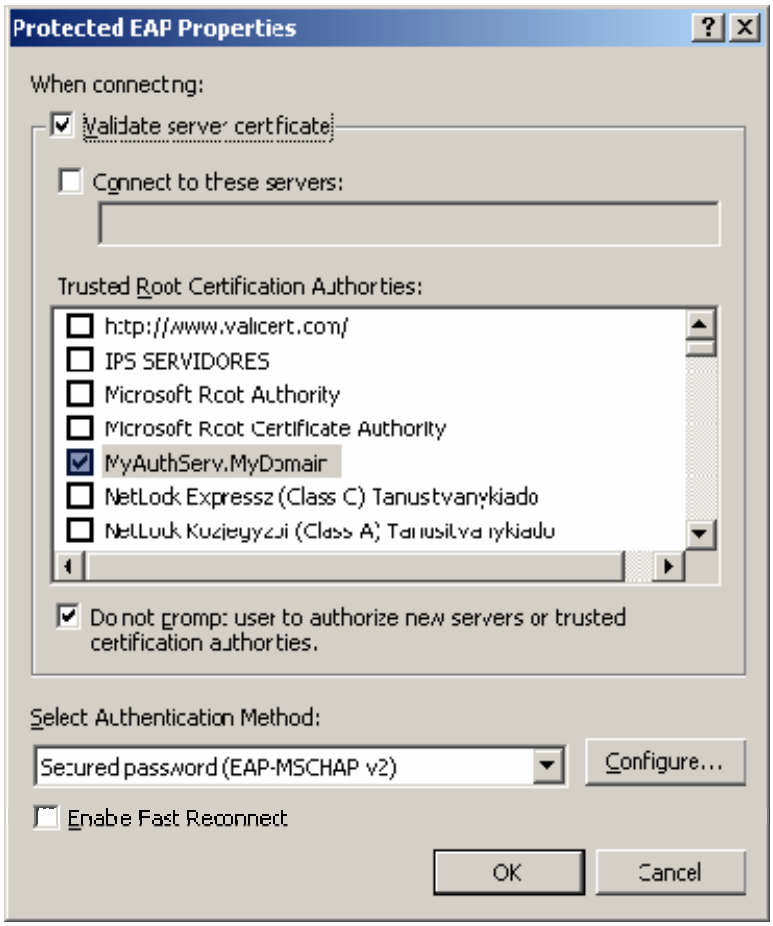
You will also need to check "Authenticate as computer when computer information is available" to enable "Machine Authentication" AKA "Computer Authentication". Machine Authentication allows your PC to connect to the network by authenticating as "Computer" before a legitimate user logs in. This allows a machine to obtain group policies just like it was connected to a wired network and this is a unique feature of the Windows Client.

If you don't have "Machine Authentication", your Group Policy will not function and non-cached users cannot log on to your machine even if they are given the proper permissions at the Domain level. "Machine Authentication" is needed to recreate the full "Wired" experience. In order for "Machine Authentication" to work, PEAP only requires that a Computer is joined to the domain. The computer will use its "Computer Password" to log on to the network.

Note that for EAP-TLS or PEAP-EAP-TLS (stronger alternatives to PEAP) to work the computer must have a "Machine Certificate" installed from the Enterprise Root CA.

Click "Properties" under "EAP type" to continue to the screen shown in **Figure BB**.

Figure BB



Protected EAP Properties

Under this section, make sure that you check "Validate server certificate" because if you don't PEAP will be as weak as EAP-FAST anonymous DH mode.

Also select the "Trusted Root Certificate Authority" for this wireless connection. Note that with SP2, it has added a new security feature with the check box for "Do not prompt user to authorize new servers or trusted certification authorities". This whole window is a very important security feature because you want your Wireless LAN to be locked by your Certificate Authority and not anyone else's.

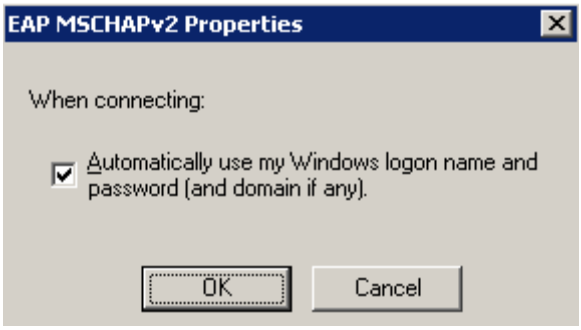
Choose "EAP-MSCHAP v2" which stands for PEAP-EAP-MSCHAPv2 mode which most people refer to as "PEAP" since this is the most common implementation. DO NOT check "Enable Fast Reconnect" since it will cause authentication problems with some Access Points like Cisco. I found this out the hard way when I had to spend time with a sniffer and Cisco tech support to figure out the problem.

If you had selected "Smart Card or other Certificate" here, it means you've set it to use PEAP-EAP-TLS mode which is a new EAP method that is suppose to be an alternative to EAP-TLS "classic" mode and maybe better but might cause compatibility problems with non-Windows Clients so I can't recommend it for now. EAP-TLS mode is configured from the previous section under "Wireless Network Properties" by choosing "Smart Card or other Certificate" instead of "Protected EAP (PEAP)". Both EAP-TLS or PEAP-EAP-TLS require client side "Machine Certificates" to work which makes them stronger two-factor authentication solutions but also much harder to

deploy. If this naming scheme is confusing to you, you're not alone. I hope this explanation clears things up for you.

Click on "Configure" to continue to the screen shown in **Figure CC**. This mode automatically logs a user on using their current Windows Credentials. If you're trying to connect to a wireless network on a Windows Domain you're not joined to, automatic logon will not work so you must uncheck this and you'll be prompted for domain credentials of the network you're attaching to. Note that it will give you the chance to save those credentials.

Figure CC



Default configuration

As you can see, this is quite a complex procedure for something as simple as PEAP (PEAP-EAP-MSCHAPv2) was designed to be. Running in either EAP-TLS or PEAP-EAP-TLS mode would make this even more complex. You definitely will want to set these policies globally using Windows Server 2003 Active Directory Group Policy.

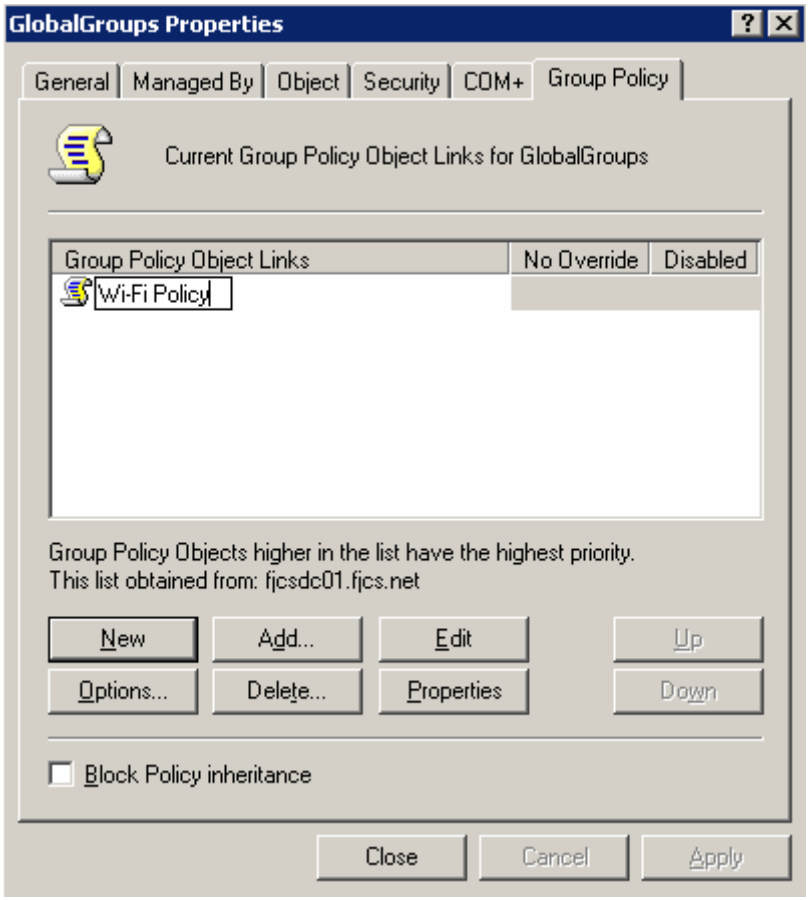
From a deployment and security standpoint, it always better to set and **enforce** all these client side settings automatically because the user is never given the opportunity to get this wrong. A wrong client side configuration can compromise your company's security. You also need to consider the expense of training your desktop support staff and the man-hours required to configure every machine in your organization using this long tedious procedure.

Automatic PEAP deployment with Microsoft Active Directory GPO

Save hours of trouble per user by deploying client side wireless configuration settings from Microsoft Active Directory with Group Policies by configuring the global Wireless LAN policy for your entire organization in minutes! These settings can be pushed out from Windows Server 2003 with Service Pack 1, Windows Server 2003 R2, or Windows Server 2007. These settings pertain to Windows XP computers with Service Pack 2 and above and they work on Windows Vista.

To get started, you need to fire up your group policy editor by opening up "Active Directory Users and Groups" as a Domain Administrator. You can deploy the wireless policy to the entire domain, or you can limit the deployment to a certain Organizational Unit (OU) that contains a certain class of users. Once you get to the screen shown in **Figure DD**, simply create a new Group Policy called "Wi-Fi Policy" and click "Edit" as shown.

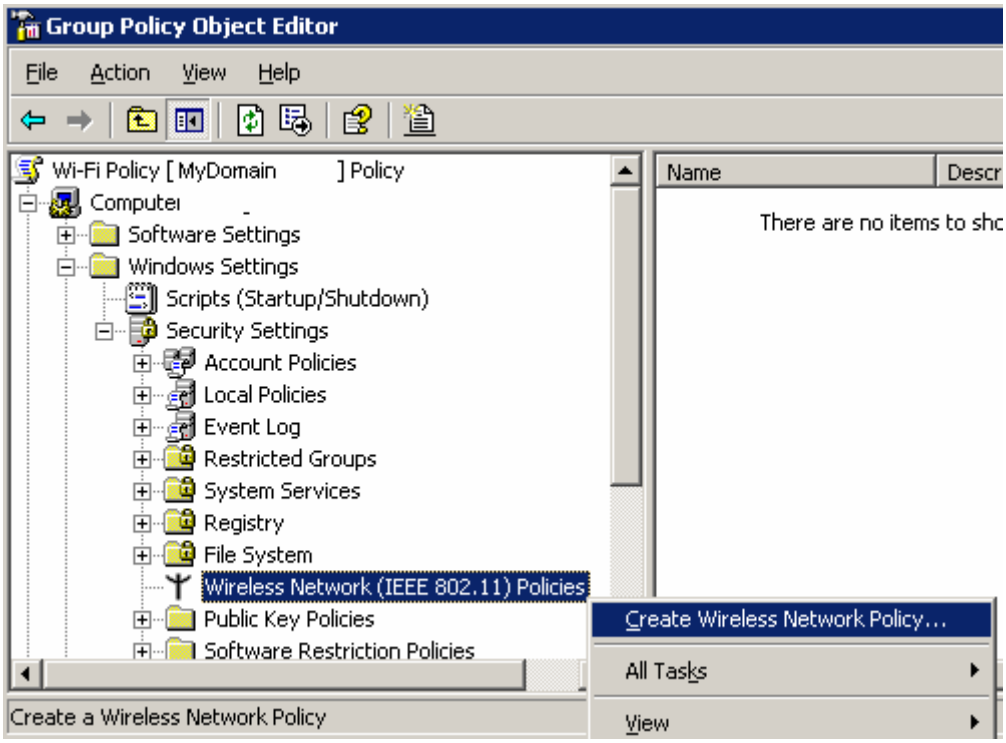
Figure DD



Edit Global Group Properties

Expand out as shown in the following screen (**Figure EE**) shot and right click on "Wireless Network ...". Click "Create Wireless Network Policy".

Figure EE



Group Policy Object Editor

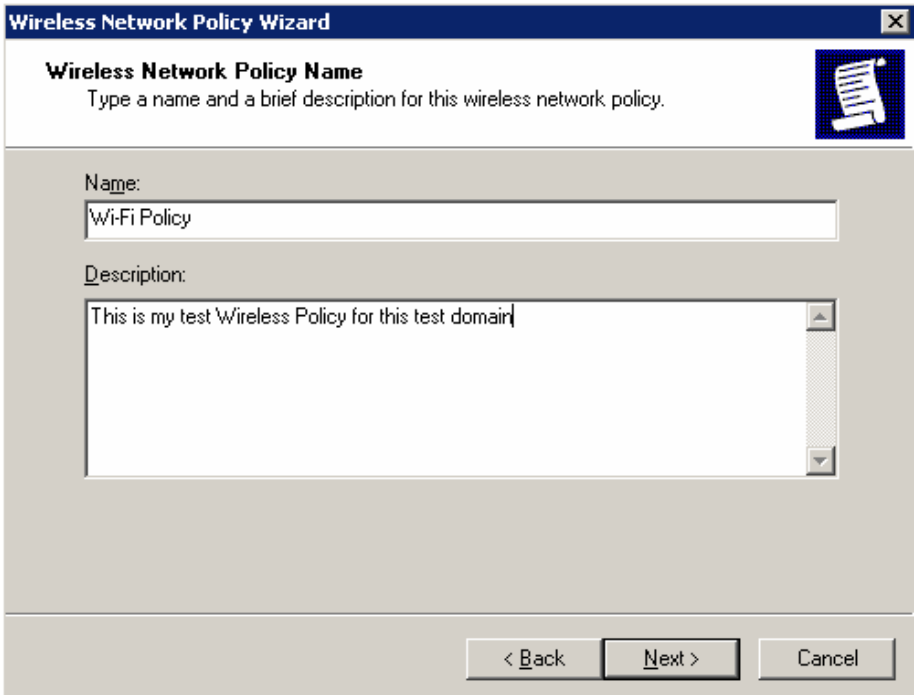
Click "Next". (**Figure FF**)

Figure FF

Wireless Network Policy Wizard

Pick a name for the new policy such as "Wi-Fi Policy" as shown below (**Figure GG**)and click "Next".

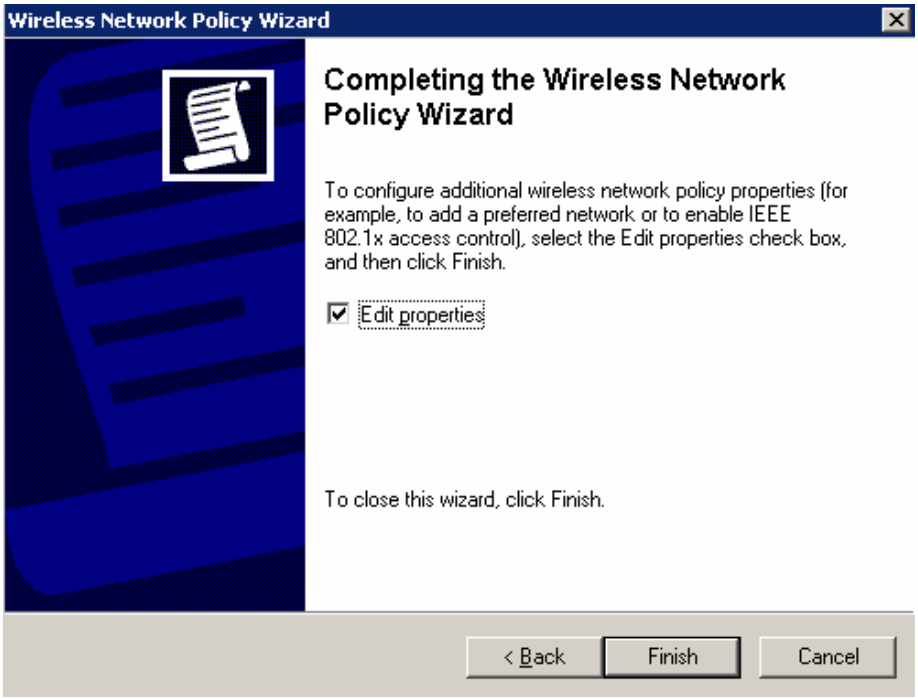
Figure GG



Choose a name

Click "Finish". (**Figure HH**)

Figure HH

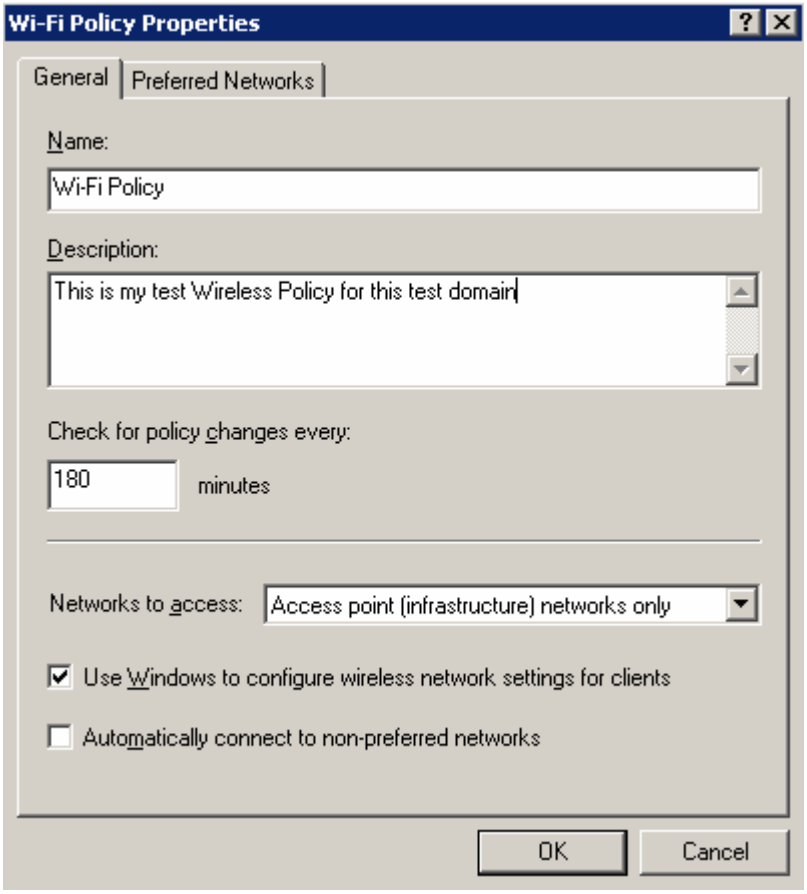


Click Finish

Fill out the dialog box shown **Figure II**. Note that I have selected "Access point (infrastructure) networks only" for "Networks to access". This is a **very important security feature** that Active Directory affords your entire network. Even if you don't intend to run wireless networking on your network, you should still use this setting to prevent any domain user from using wireless ADHOC mode.

The "Use Windows to configure wireless network settings for clients" is also a very important feature. Even if your clients have a third party wireless client like the Cisco ACU installed with Microsoft Wireless Zero Configuration disabled, this setting will override all of them and make them use these Active Directory settings. This ability allows you full control of wireless networking in your network from a centralized policy. When you've completed this page, click on the "Preferred Networks" tab

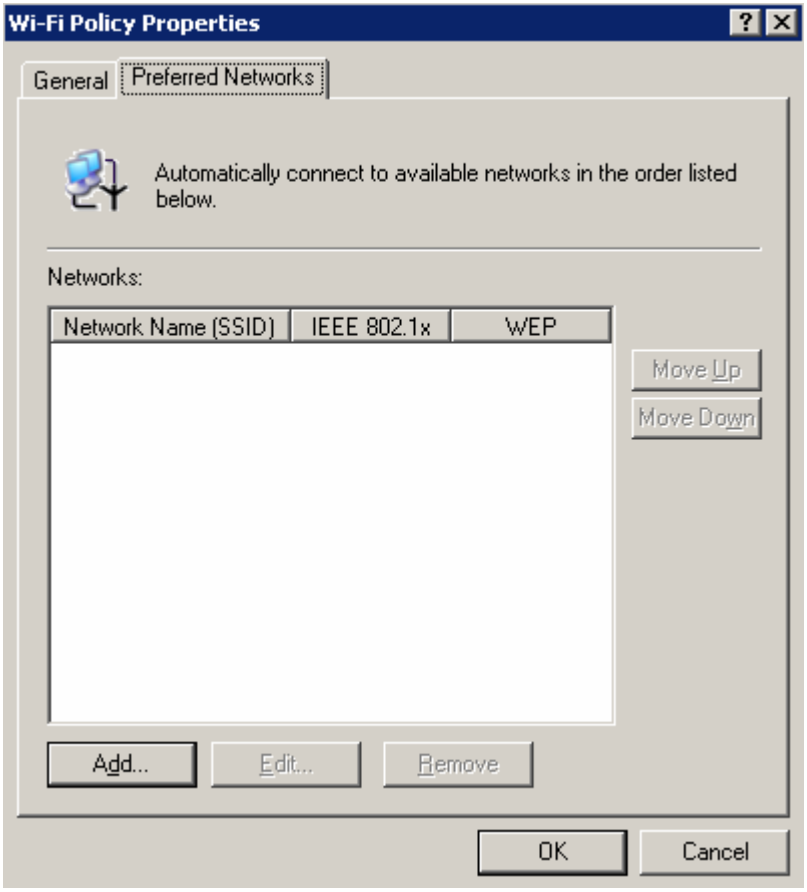
Figure II



Wi-Fi Policy

Click "Add." (**Figure JJ**)

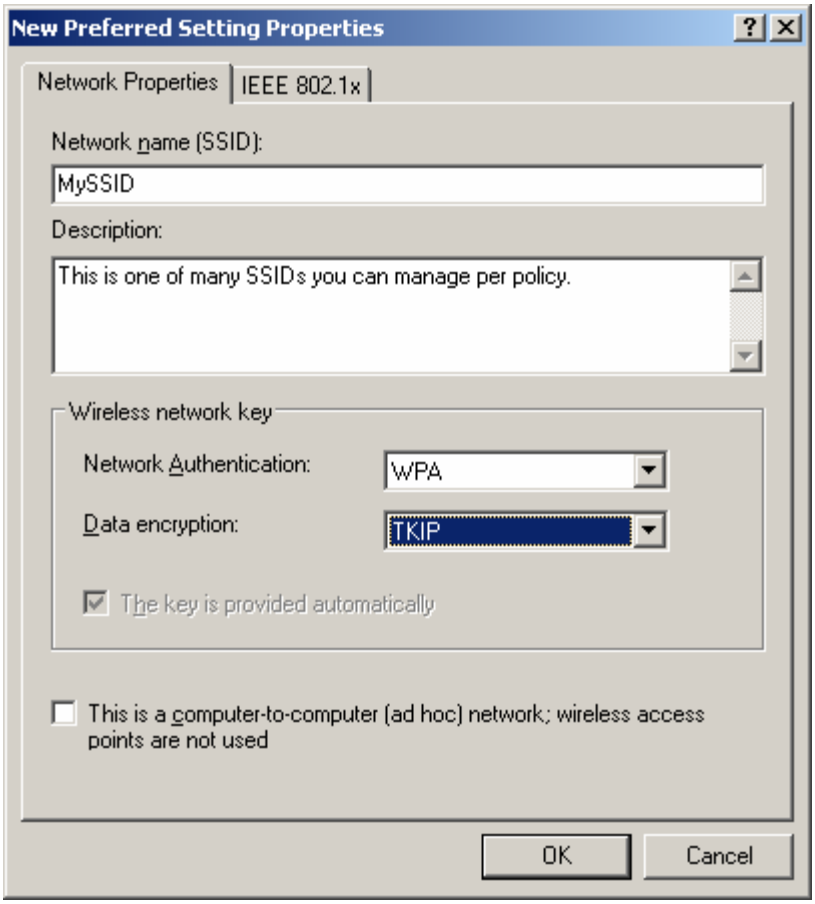
Figure JJ



Preferred Networks

Put in the SSID you have configured for all of the Wireless Access Points you have set up. Use the SSID that you want, it shouldn't be the same as my example below. Having a common SSID across all your access points (if they attach to the same subnet) allows your users to "roam" between access points. Fill everything else out below the same way with WPA and TKIP as fairly secure and widely supported settings. You can use AES for "Data encryption" if all of your hardware supports it. Then click on the "IEEE 802.1x" tab. (**Figure KK**)

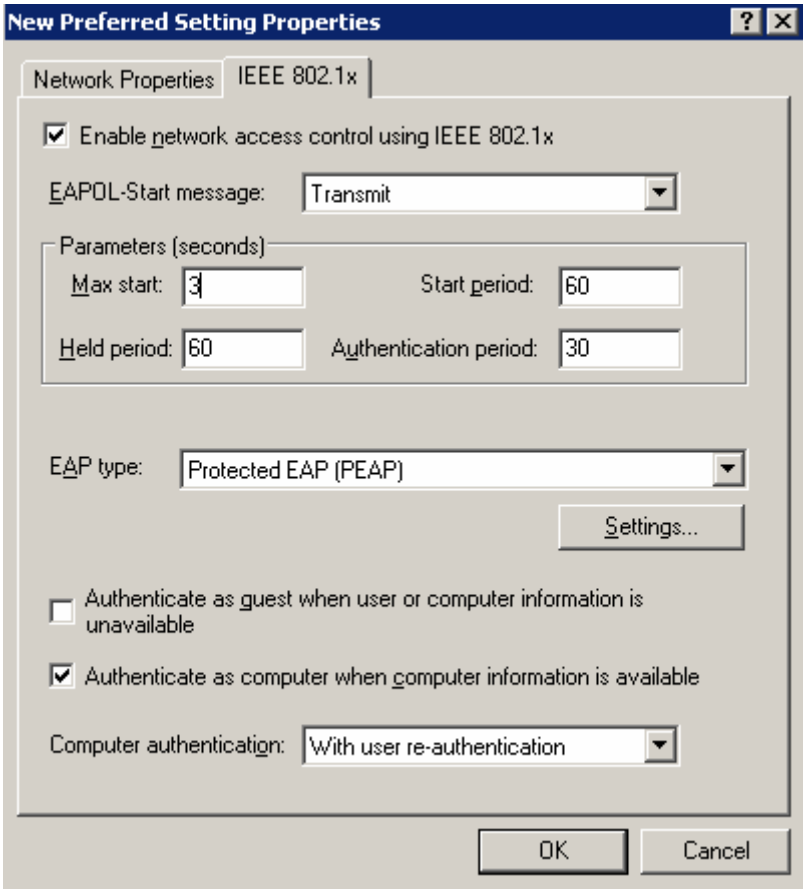
Figure KK



MySSID

Fill out the following window exactly as shown in **Figure LL**. The "Authenticate as computer when computer information is available" enables machine authentication which is a very useful and unique feature of the Windows wireless client. Click on the "Settings" button to continue.

Figure LL

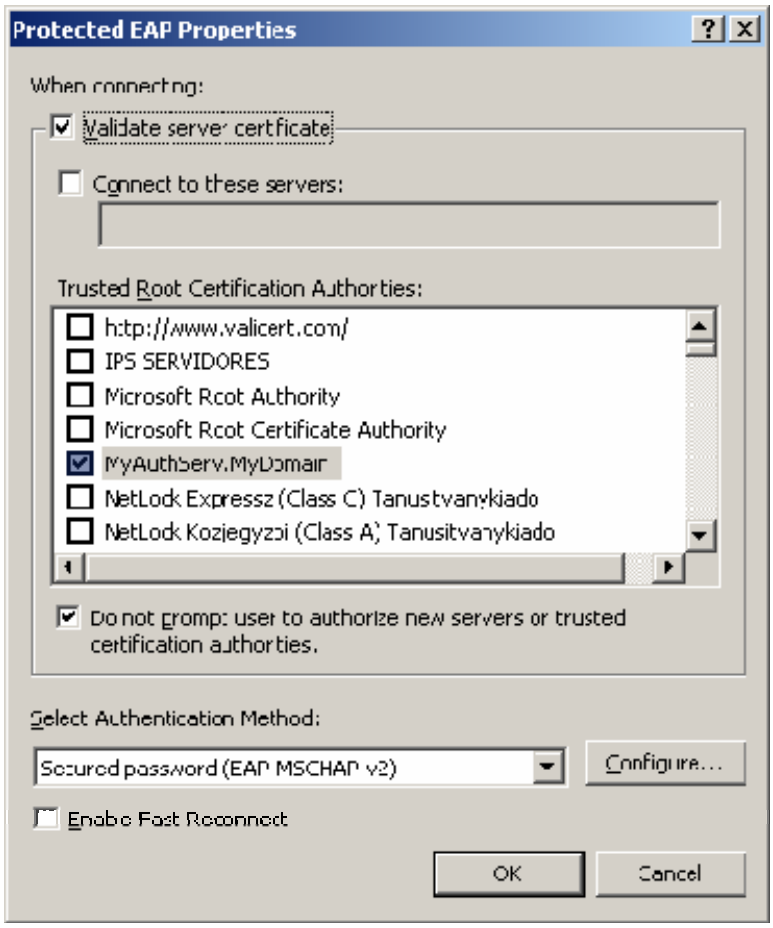


Preferred settings

On the following screen as shown in **Figure MM**, is another important security feature of Active Directory. Forcing the "Validate server certificate" setting protects users from Man-in-the-Middle attacks where a hacker may attempt to pose as a valid access point and RADIUS server with a bogus certificate. Validating the server certificate prevents this and you don't want the users to set this themselves because that means they have the opportunity to screw it up.

Pre-selecting the "Trusted Root Certificate Authorities" is also an important security feature because you don't want the users to trust all of the other Certification Authorities on the list. You also want to check "Do not prompt user to authorize new servers or trusted certification authorities. Manual configuration of these settings is complex and unreliable. Security must be systematically enforced at a policy level.

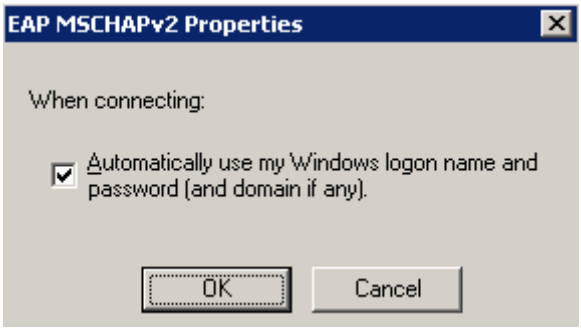
Figure MM



Certificate settings

Then click on the "Configure" button in **Figure MM**, and you will get the Window shown in **Figure NN**. This is a very convenient method of authentication where the user's Windows credentials are automatically used for Wireless (or Wired 802.1x mode) access.

Figure NN



Automate credentials

Once this is completed, hit OK and OK all the previous dialog boxes to commit the changes. Within minutes or an hour once your client machines refresh their group policy automatically or when they logon to the Active Directory, they will have all these new settings.

Microsoft IAS RADIUS for wireless authentication

Windows Server 2003 comes bundled with a very capable RADIUS (also known as AAA) server that's extremely stable, secure, and robust. When you search on Internet security databases for Microsoft IAS vulnerabilities, you won't find any. The IAS service just runs for years without the need to patch IAS. If your Windows Server 2003 box is hardened to only accept IAS requests with host-based firewall restrictions on all other ports and you install no other services on a Windows 2003 box, you can literally keep an IAS RADIUS server up for years of zero downtime or reboots.

IAS competitors

One of IAS' biggest competitors in the Enterprise market is Cisco ACS which people often assume they must use if they're using Cisco networking equipment which simply isn't true. Your Cisco network equipment works perfectly fine so long as you avoid proprietary, less-secure harder-to-deploy protocols, like LEAP or EAP-FAST.

Furthermore, the stability of ACS is questionable and there is an endless patch cycle for it since it has been plagued with security vulnerabilities and bugs. I've spent my share of time troubleshooting ACS and many hours on tech support. I have had a lot of hands-on experience with Cisco ACS. The latest version of Cisco ACS 4.x currently has two unpatched security vulnerabilities one of which is **critical**. Version 3.x and 2.x also have their share of critical vulnerabilities some of which are unpatched as of December 10, 2006.

Cisco ACS also lacks the ability to act as a relay RADIUS server which limits its ability to serve in a more robust multi-tier RADIUS environment. You need that ability to link to multiple Active Directories or other user directories that are not tied to each other. ACS also costs around $8000 per copy whereas Microsoft IAS comes with Windows Server 2003. Two redundant RADIUS servers would add up pretty quickly. Cisco ACS also comes on a dedicated appliance but that's even harder to use in my experience since you don't even get a Windows console graphical interface to work with.

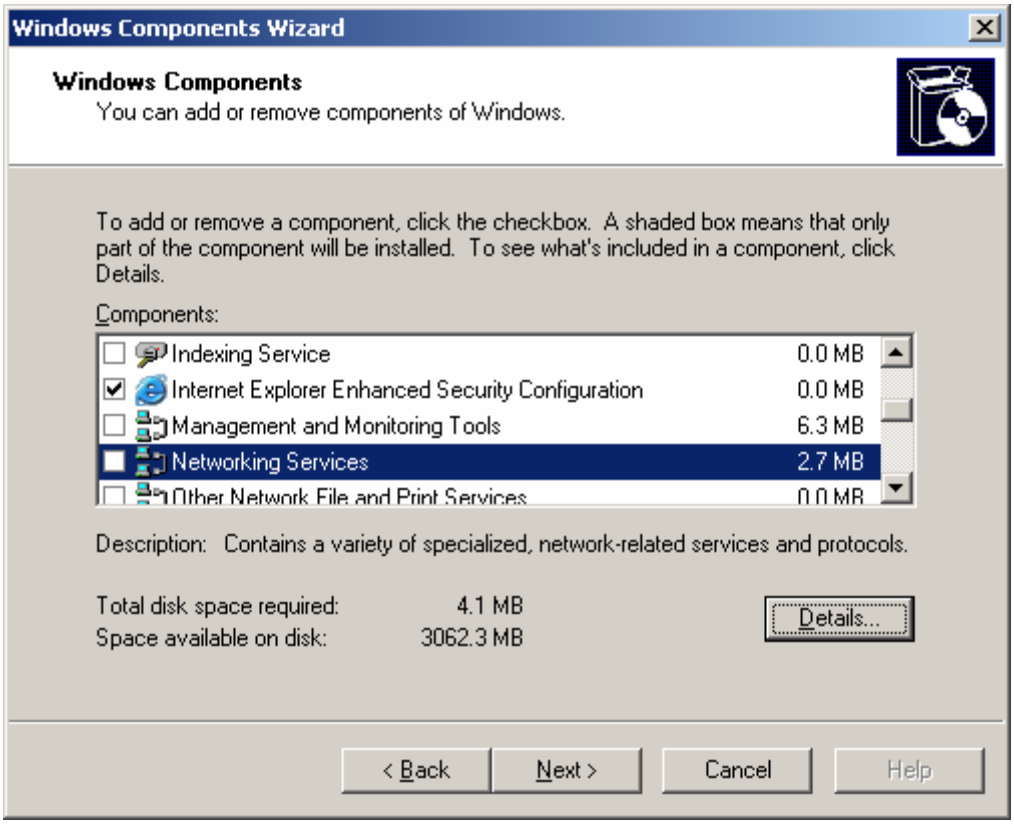
Funk software (acquired by Juniper) has a pretty good solution with Steel-belted RADIUS at around $4000 per copy but that is still a significant cost especially when you need two RADIUS servers for redundancy. Funk is a great solution for companies which don't run a Windows Active Directory environment because IAS is tightly wound in to Microsoft Active Directory and doesn't support non-Microsoft databases.

Linux users have FreeRADIUS available to them. FreeRADIUS has had critical flaws (0.x and 1.x) in the past but they're all patched now unlike Cisco. FreeRADIUS still isn't as clean as the Funk or Microsoft RADIUS solutions but it's completely free if you're rolling your own Linux distribution or you don't need enterprise Linux support. If you're talking about SuSE or Red Hat and you want enterprise support, then the annual support contracts is double the cost of a perpetual Windows Server 2003 license. It all depends on usage model and some people will prefer Linux and some will prefer Microsoft.

Install IAS

Windows Server 2003 doesn't come with any extra services installed by default for security reasons so you'll need to manually install IAS. It's fairly simple if you have the Windows Server 2003 install CD. To install IAS, simply open "Add remove programs" from your control panel and select "Add remove windows components". You will see the following window (**Figure OO**) so you'll need to scroll down to "Network Services". You don't want to just check it because you don't want all Network Services installed, just highlight it and hit the "Details" button.

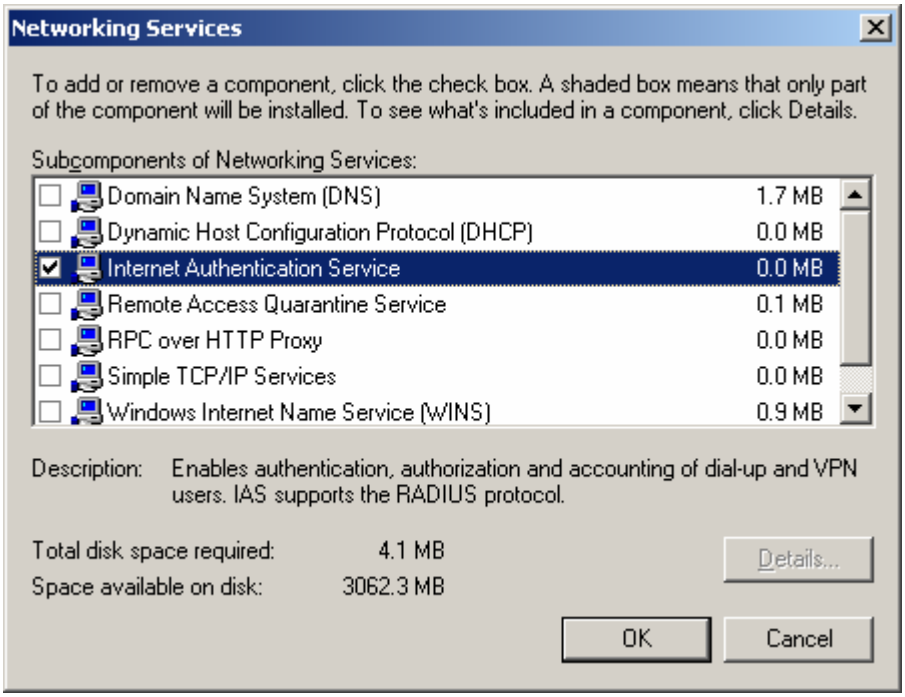
Figure OO



Networking services

Once you get to the screen shown in **Figure PP**, scroll down and just check off "Internet Authentication Service" IAS for short.

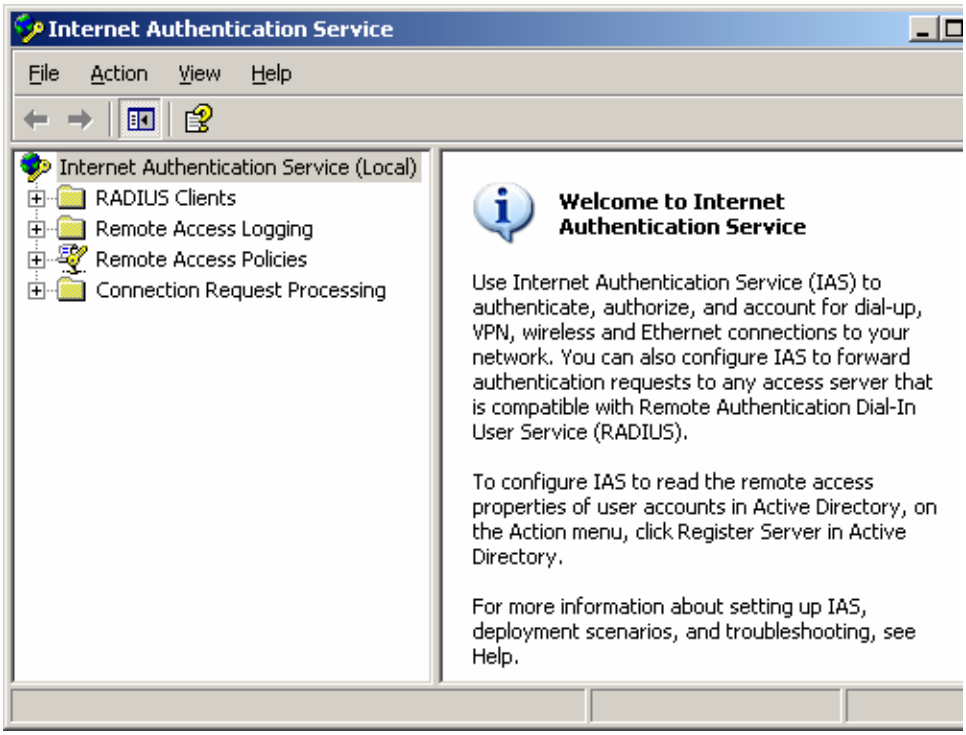
Figure PP



Internet Authentication

Once you've installed IAS, you'll be able to launch IAS from your Administrative Tools either from the control panel or from the start menu. You'll see the following interface. (Figure QQ)

Figure QQ

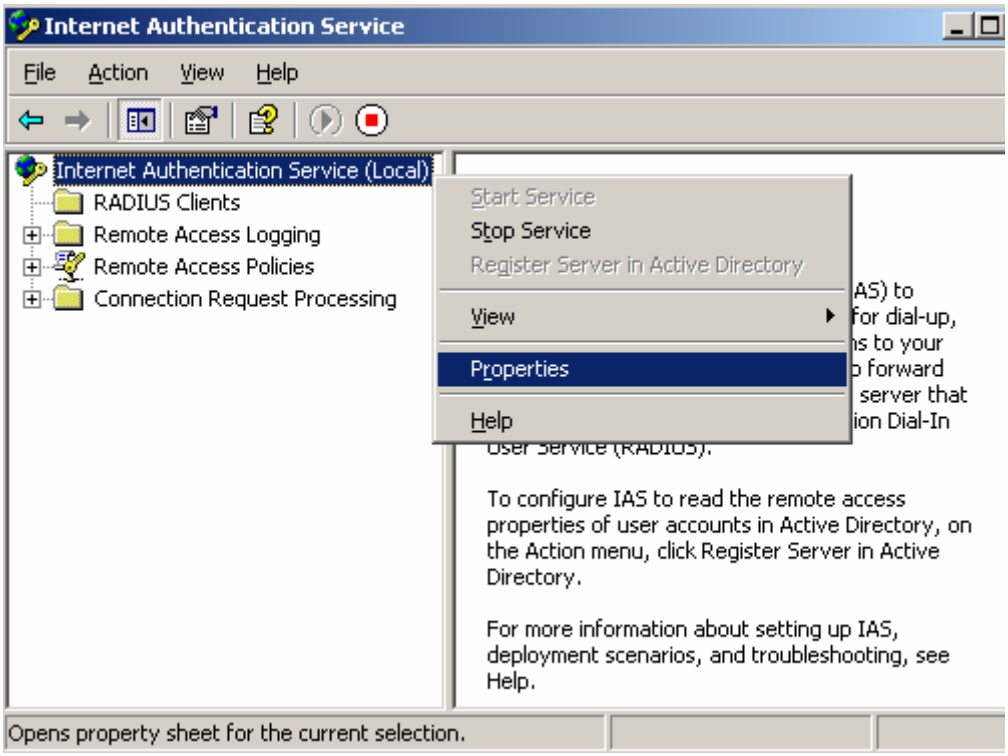


Service

Set logging policies

The first thing we'll do is look at and set the logging policies (**Figure RR**). Right click on "Internet Authentication Service (Local)" and click Properties.

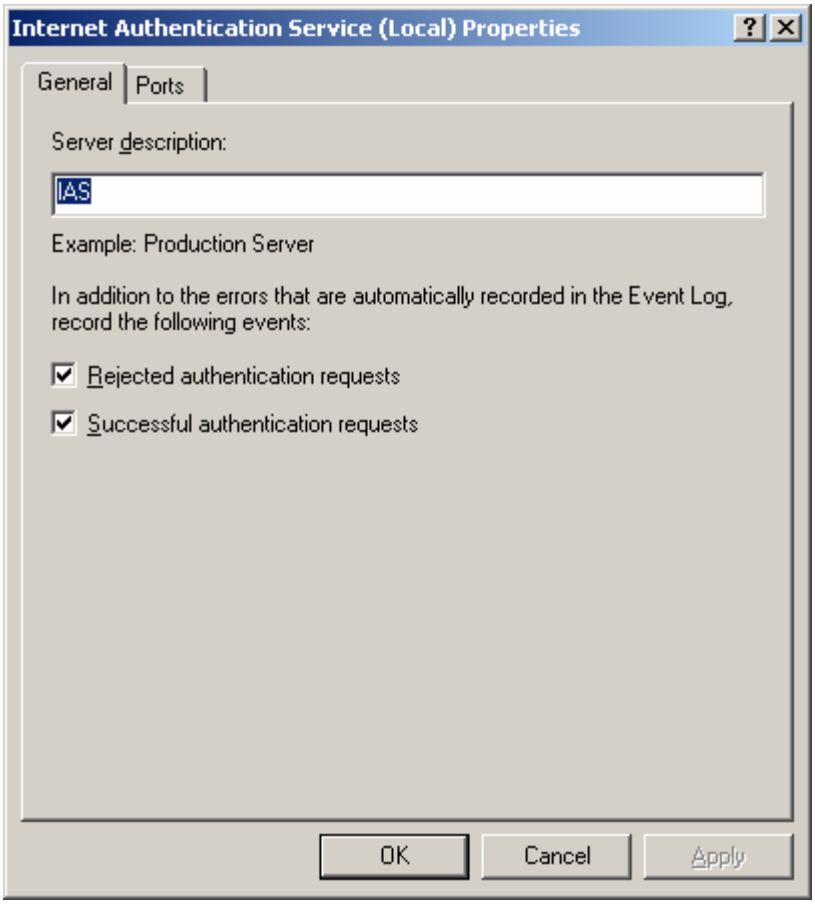
Figure RR



IAS Properties

You will now see the screen in **Figure SS**. If you check off the two checkmarks here, you will force IAS to log successful and reject authentication requests to the Windows Event Viewer. If you're intending to use text or SQL based logging, you don't need to check these unless you want the logs showing up in both places.

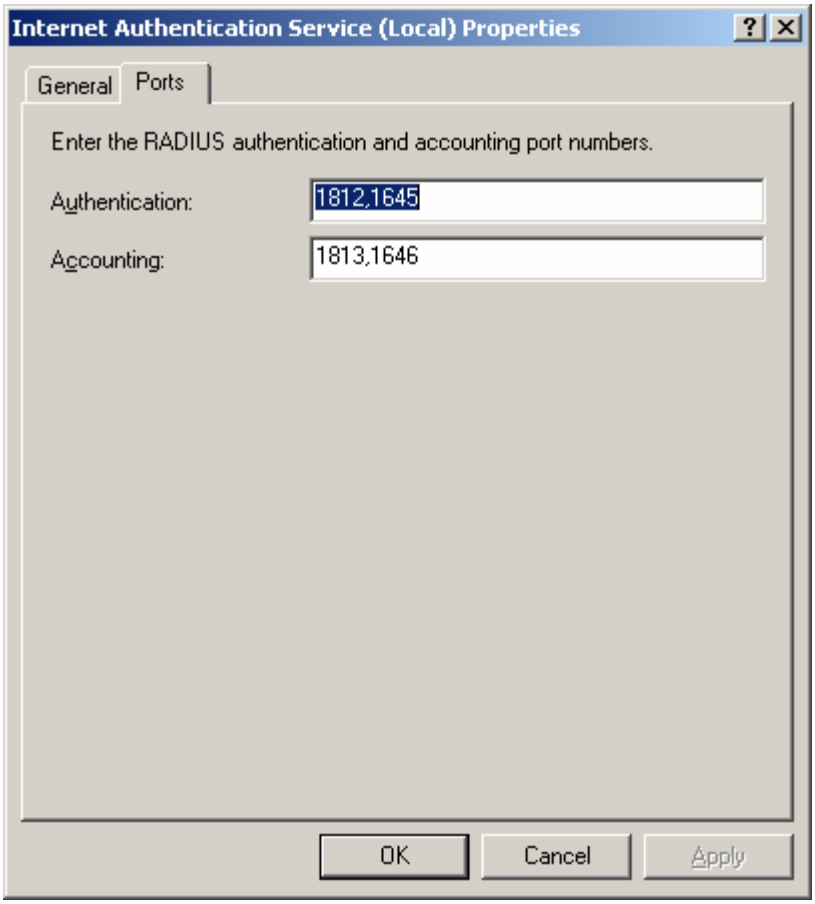
Figure SS



Local Properties

If you click on the "Ports" tab, you'll see the screen shown in **Listing TT**. These are the default RADIUS ports and you should generally leave them alone for standardized RADIUS conventions. Microsoft IAS will actually listen on two sets of ports. The lower number ports are the more traditional port numbers, Microsoft applications prefer the higher number ports. You should generally leave this setting alone as is.

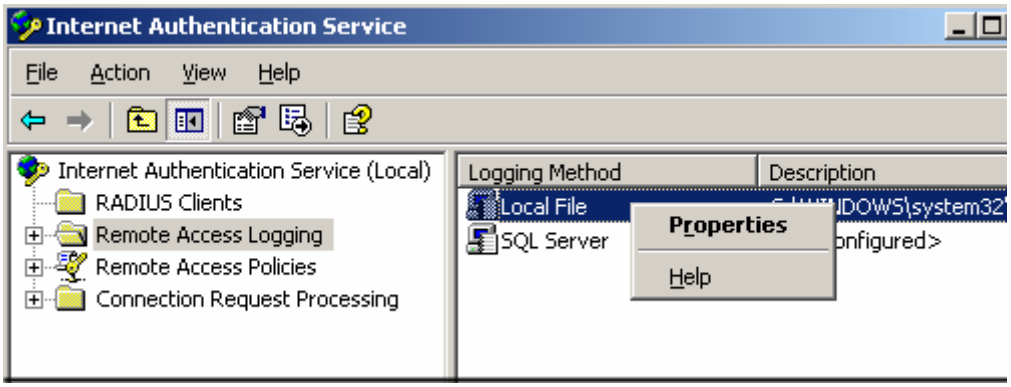
Listing TT



Ports

This next part lets you set the standalone text and SQL logging capability of Microsoft IAS. You right click on "Local File" under "Remote Access Logging" page and hit "Properties". (**Figure UU**)

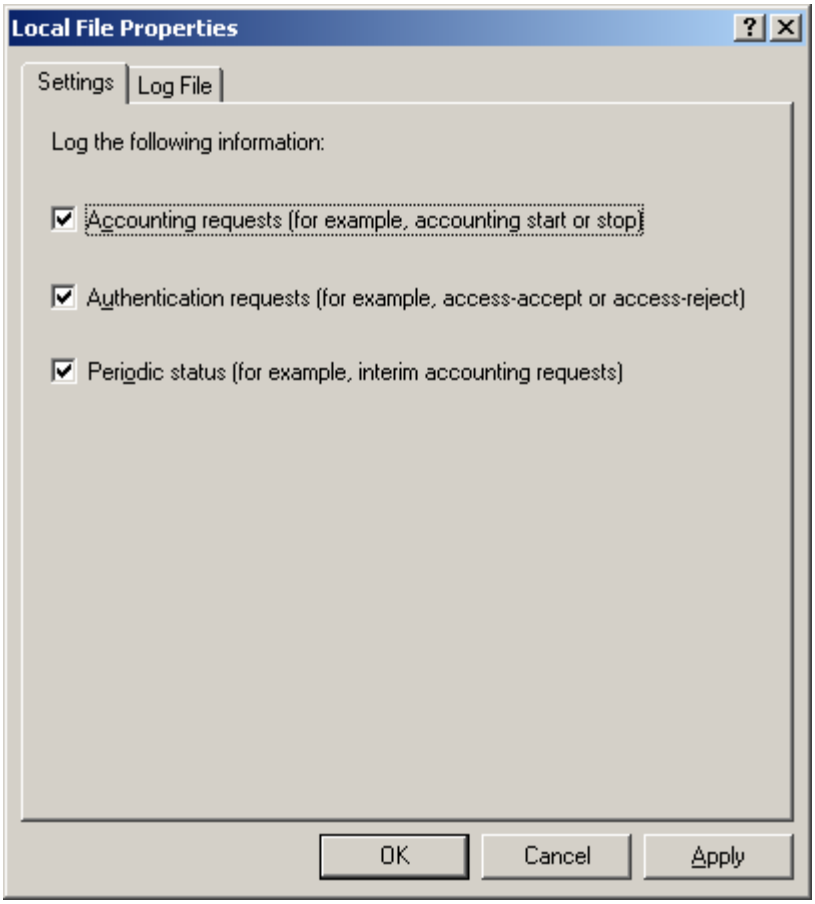
Figure UU



Remote Access Logging

The first "settings" tab lets you set what events you want to log. (**Figure VV**)

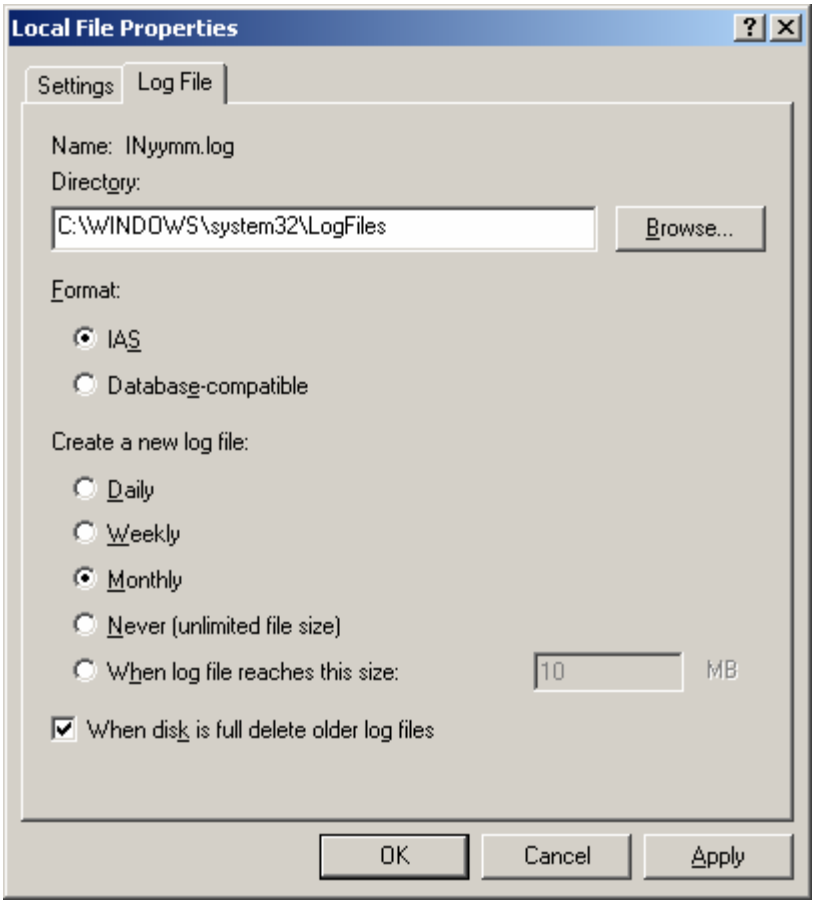
Figure VV



Local file properties

The "Log File" tab lets you set the file format and the log size limits. (**Figure WW**)

Figure WW



Log file

We won't go in to SQL logging in this article because it gets rather complex to set up a SQL database. You have to manually create the accounts and tables in SQL in order for this to work. Furthermore, IAS under Windows Server 2003 insists on stopping the RADIUS service if logging doesn't work so if the SQL server doesn't respond, all of your RADIUS servers stop working. Unfortunately, Microsoft doesn't give you any way to override this "fail-shut" behavior because they claim customers want it this way because it's more secure but every customer I've talked to wants a choice on this behavior.

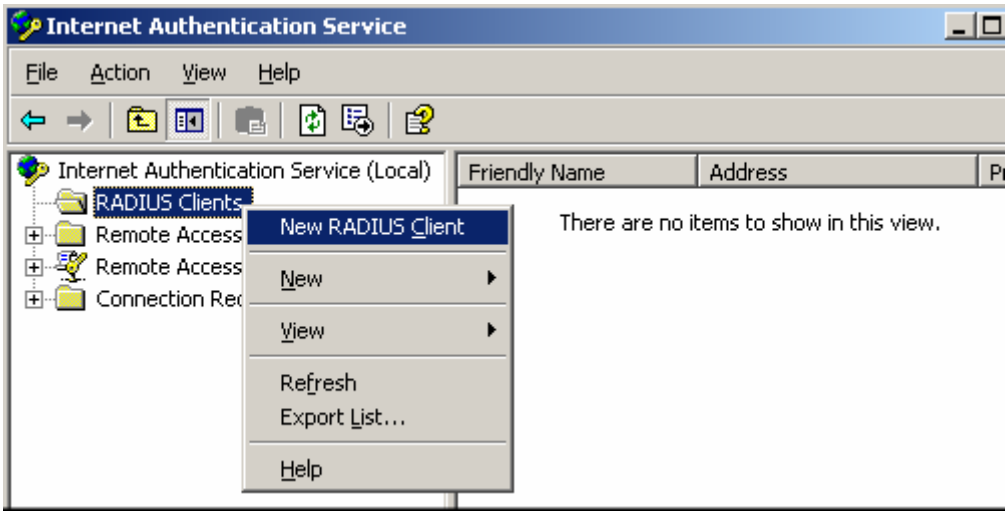
There is no security risk because the Authentication and Authorization component of Microsoft IAS is working perfectly fine, it's merely unable to make a record of the transaction. I've spoken with Microsoft and they're telling me they will correct this with Windows Server 2007 (or whatever it's going to be called when it's released next year). Hopefully they'll automate the SQL database creation process with a script too.

Add RADIUS clients

A RADIUS "client" is not what you would typically think of as a "client" as in a user. A RADIUS client is something like a wireless access point, a router, a switch, a firewall, or a VPN concentrator. Any device that provides network access that needs to delegate AAA (Access, Authorization, and Accounting) to a RADIUS server is considered a RADIUS client. For the purpose of this tutorial, we'll set up a single access point as a client.

To start, we'll right click on "RADIUS Clients" and select "New RADIUS Client" as shown in **Figure XX**.

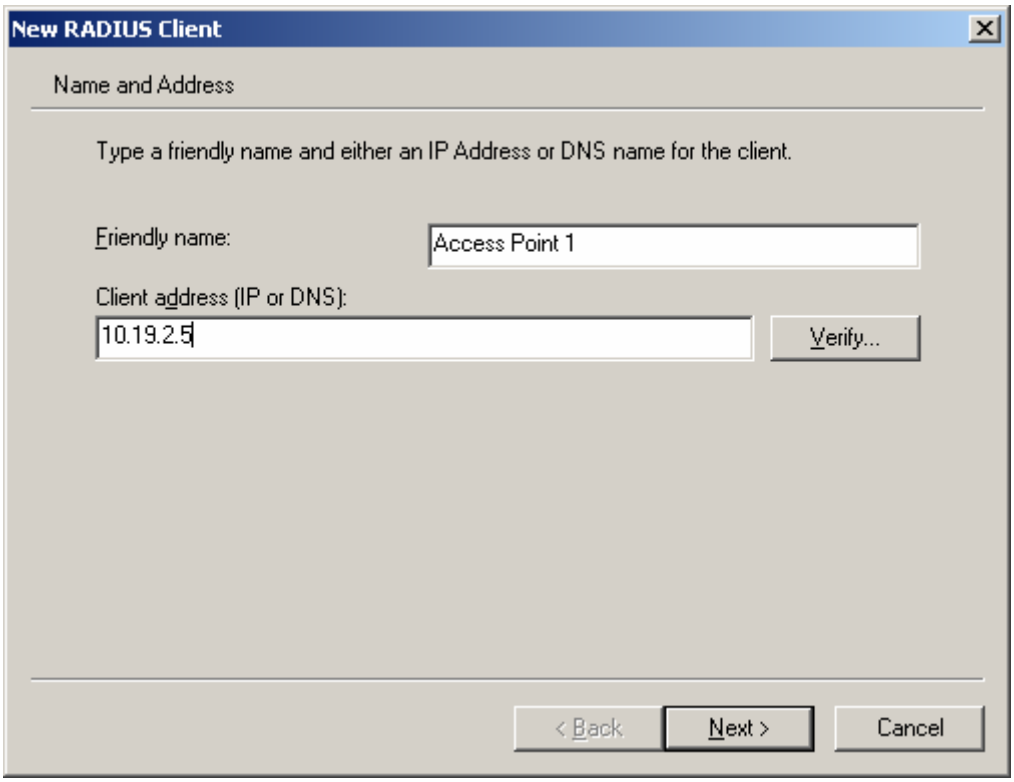
Figure XX



Radius Client

You then get the screen shown in Figure YY where we give the device its name and set the IP address of the access device which in this case is an access point. Be aware that if you're talking about a router or firewall that has multiple IP addresses because it has multiple interfaces, you must configure the IP address that is closes to the RADIUS server. This is because the RADIUS request is seen as coming from the closest interface on a multi-homed access device and if you configure the wrong IP, it will not be able to communicate with the RADIUS server.

Figure YY

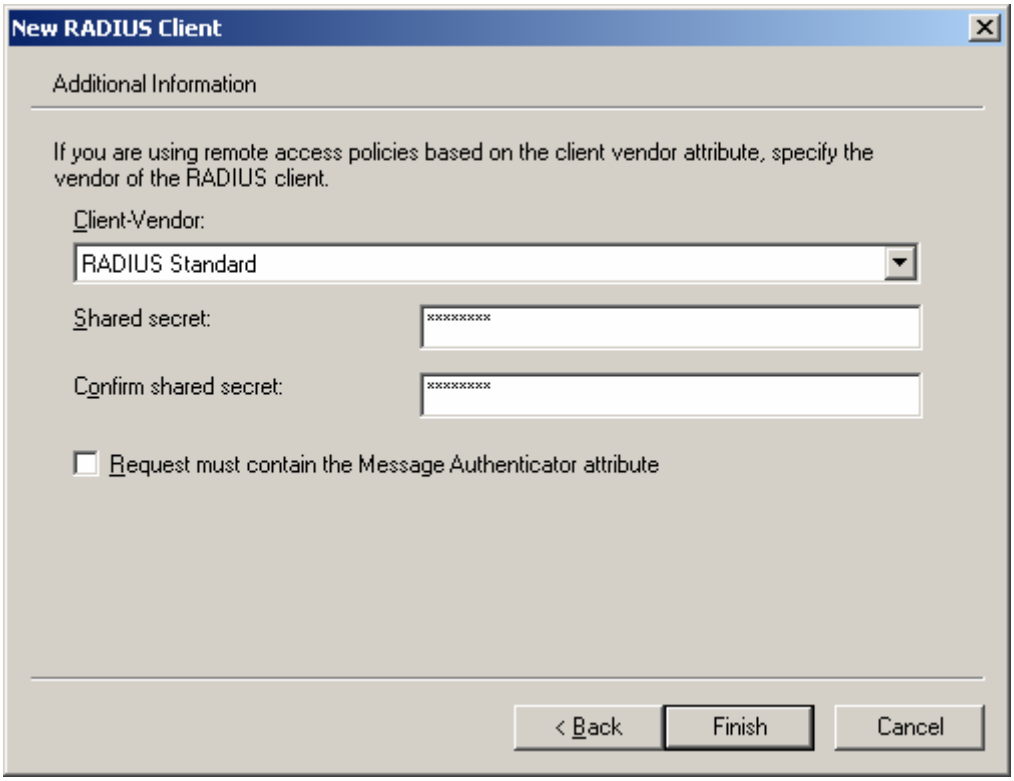


New RADIUS client name and IP

Then we set the RADIUS type and RADIUS secret. The RADIUS type is almost always set to "RADIUS Standard". Cisco devices are the exception and you must select "Cisco" for the "Client-Vendor" field if you want your Cisco devices to work. There are exceptions like Cisco wireless switches because the switches were acquired from Airespace in 2005.

Airespace wireless switches use "RADIUS Standard" like everyone else in the industry. The "shared secret" is the secret shared between the RADIUS server and the access device (**Figure ZZ**). Try to make the secret 10 characters or more comprised of random numbers and letters. Avoid spaces and special characters since that might have incompatibilities in some devices and software and you'll have a rough time troubleshooting.

Figure ZZ



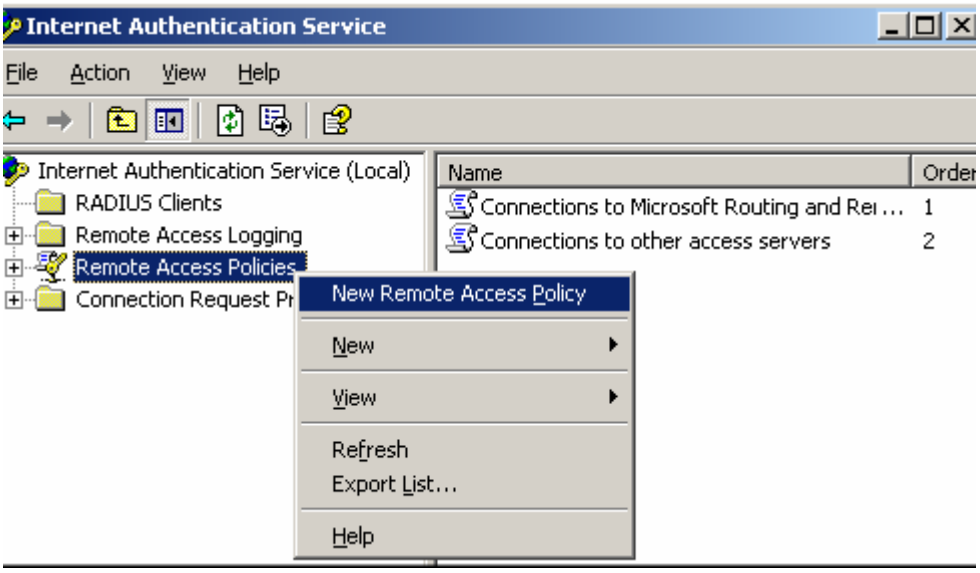
Setting the shared secret

Click "Finish" to complete. You'll need to repeat this for all of your access devices.

Add remote access policies

Now we need to create a remote access policy to authenticate and authorize the user trying to access our access devices. To do this, right click on "Remote Access Policies" and click "New Remote Access Policy". (**Figure AAA**)

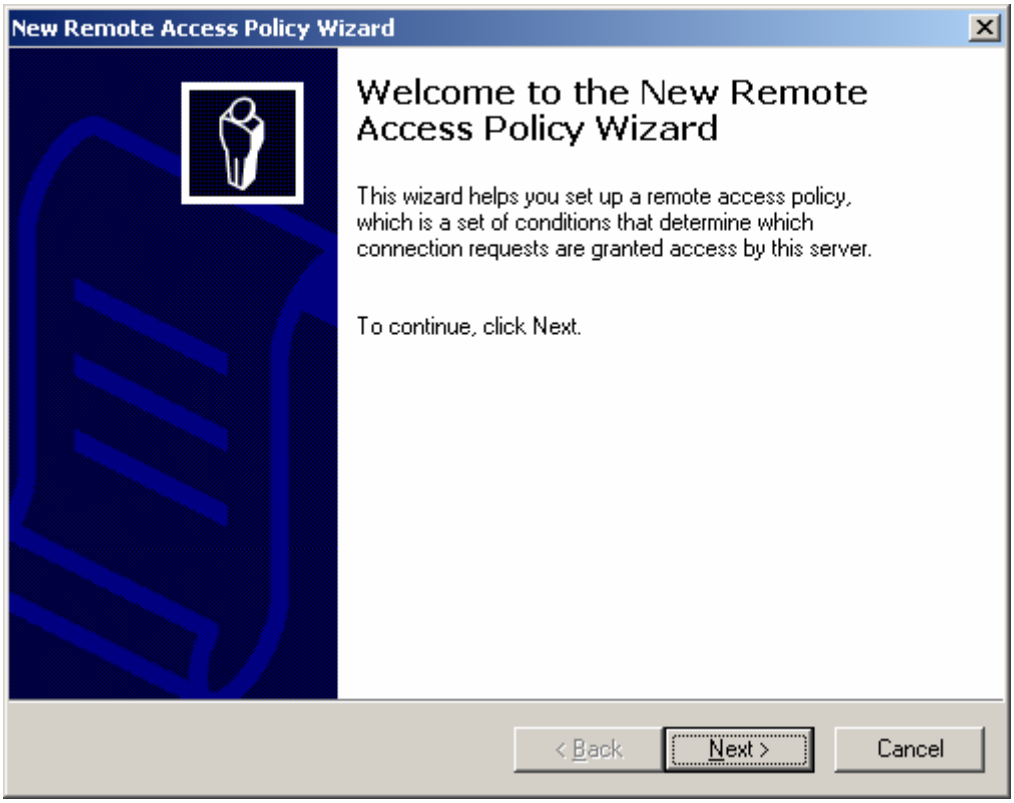
Figure AAA



New Remote Access Policy

Click "Next" to move to the next screen (**Figure BBB**).

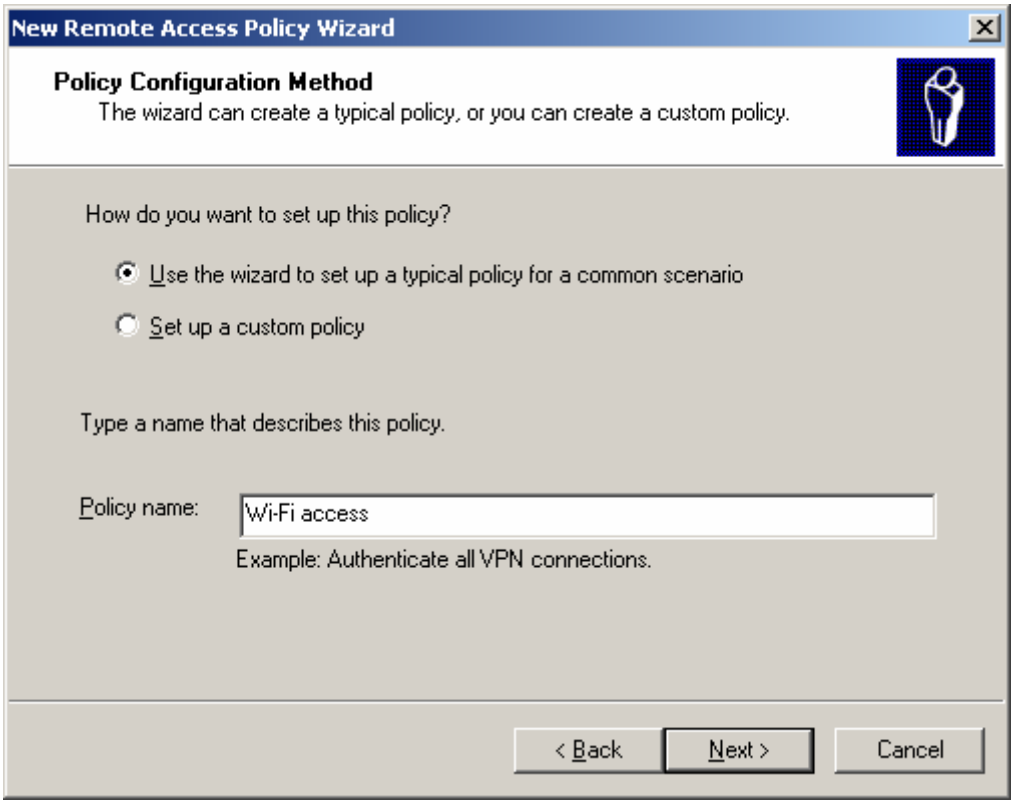
Figure BBB



Policy Wizard

Give your policy a name and use the wizard. Hit "Next". (**Figure CCC**)

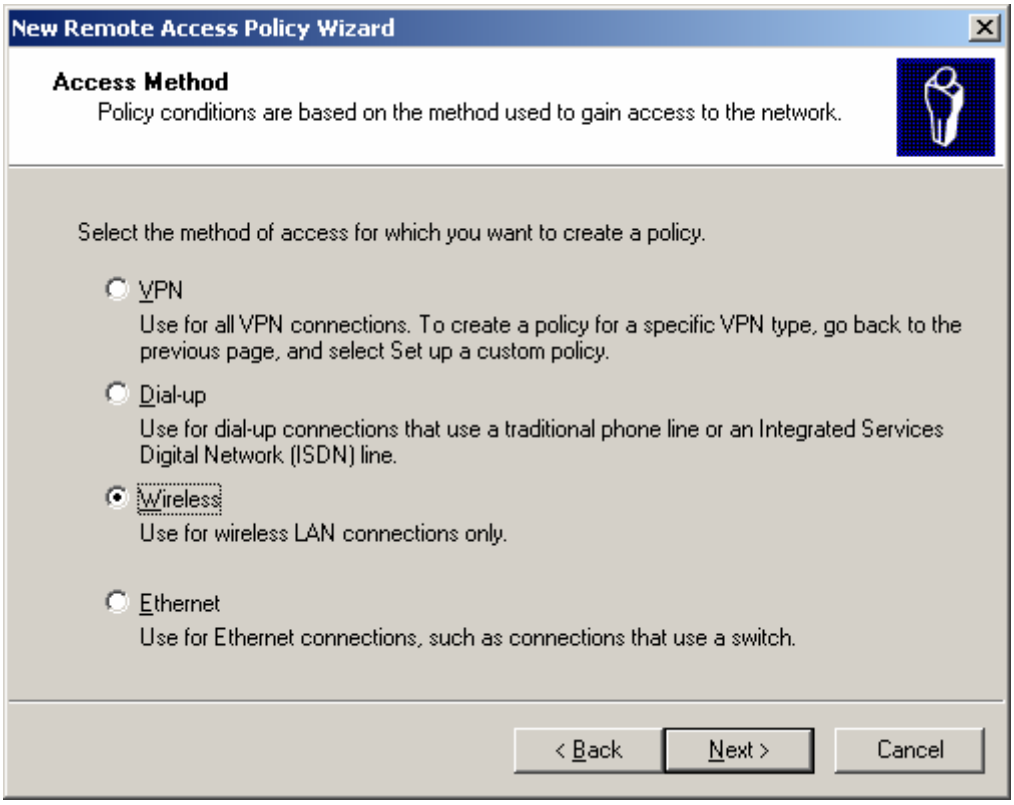
Figure CCC



Policy Name

Choose "Wireless" and hit "Next". (**Figure DDD**)

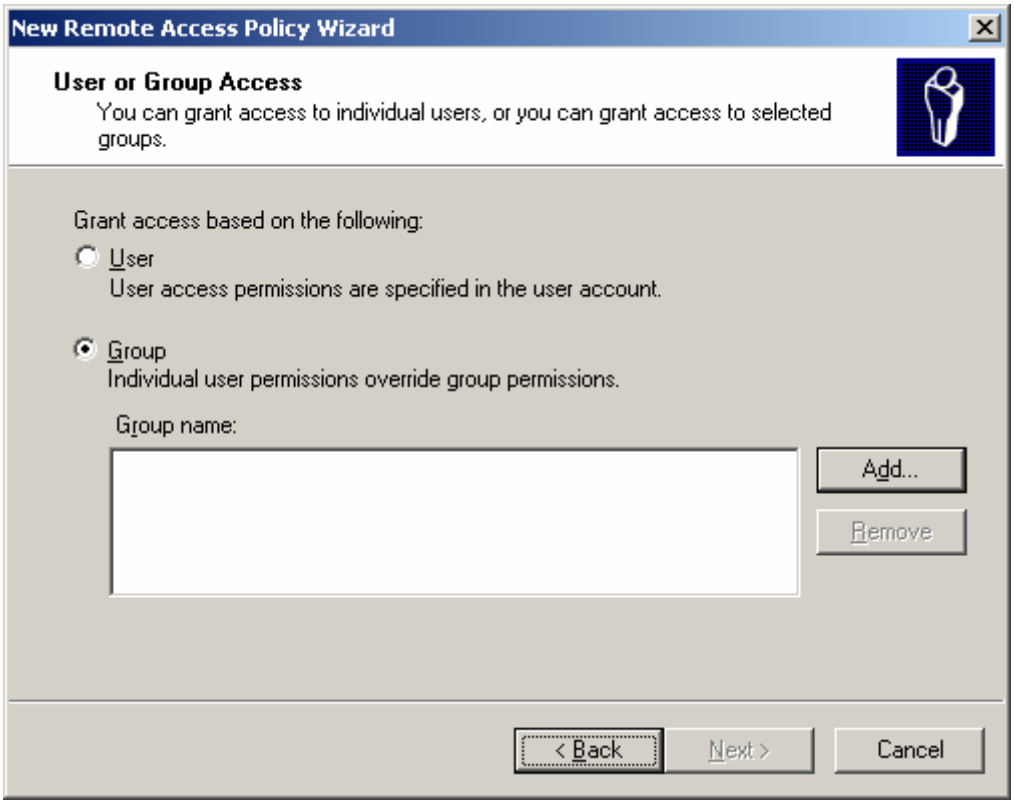
Figure DDD



Wireless

Here you'll need to grant access to your users and computers. Hit "Add". (**Figure EEE**)

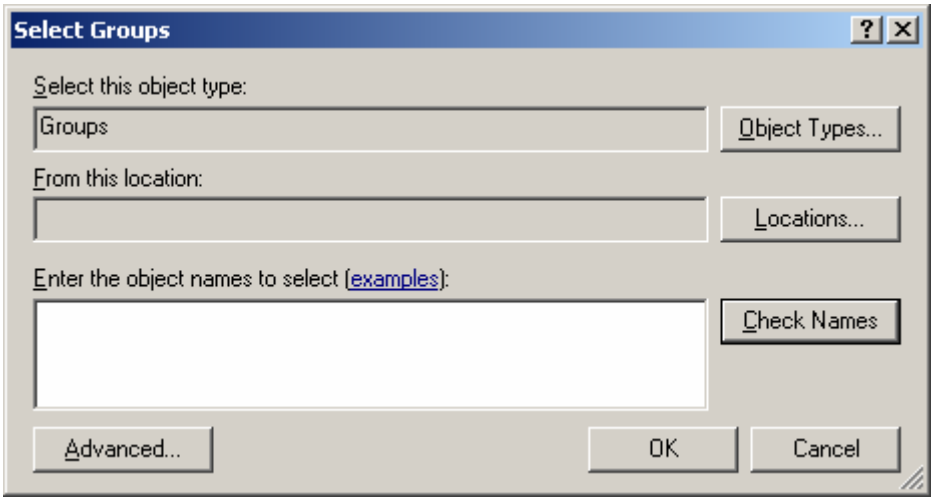
Figure EEE



Group Access

Here you'll need to adjust the location to your domain. Hit "Locations". (**Figure FFF**)

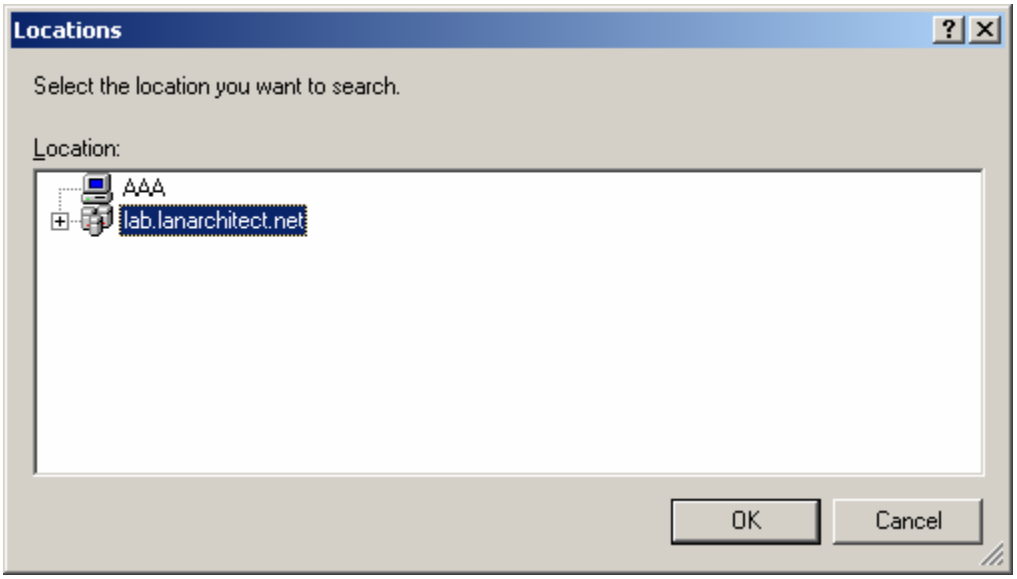
Figure FFF



Select Groups

Choose the domain you're trying to authenticate to and hit "Ok". Note that the IAS server must be joined to the domain you're authenticating to or a trusted domain. (**Figure GGG**)

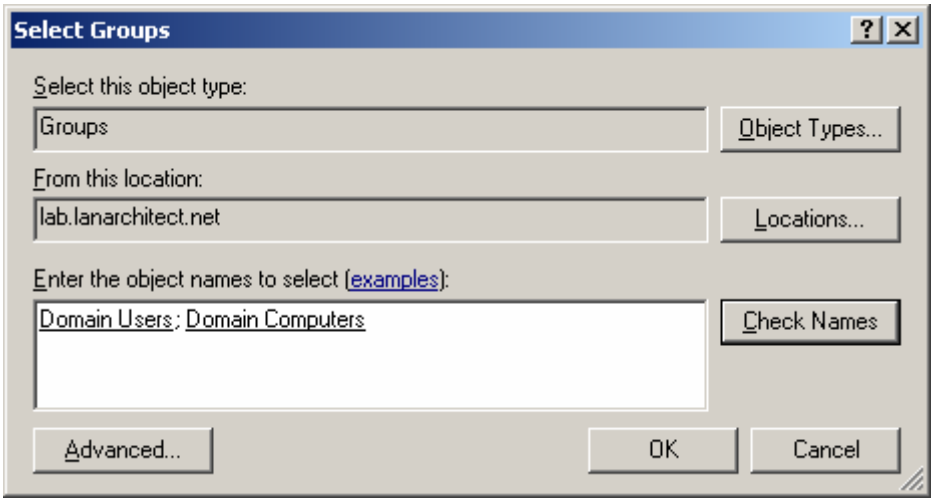
Figure GGG



Select Location

Type "Domain Users" and "Domain Computers and separate them with a semicolon. (**Figure HHH**) Then click on "Check Names" to force it to underline and validate your entries. You may of course restrict access to a smaller group of users and computers since the following option allows all domain users and all domain computers to connect to your wireless LAN. Hit "Ok".

Figure HHH



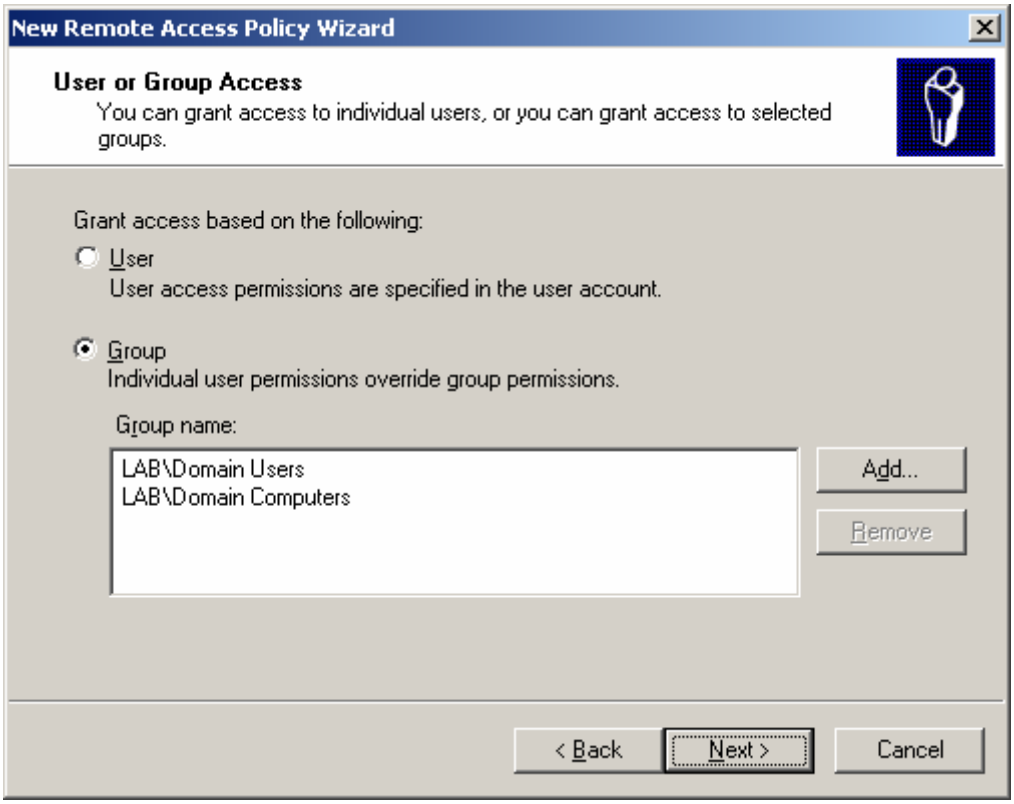
Enter domain

Note that "Domain Computers" is used to authenticate your computer for "machine authentication" which connects your wireless PC before the user even logs in. This is a very useful and unique benefit of the Windows Wireless Client since it emulates the full wired experience for wireless users.

If "machine authentication" isn't implemented, group policies and login scripts won't fire off. Furthermore, only cached users can login to the wireless computer, because users who have never signed on to that PC can't authenticate with the domain. For this reason alone is enough for me to always recommend using the Windows Wireless client for Windows users not to mention the auto-deployment capability.

Now you see the screen shown in **Figure III** with a summary of the user and computer groups you're allowing access. Note that this is an OR operator between these two group names. Either one true registers a success. Hit "Next".

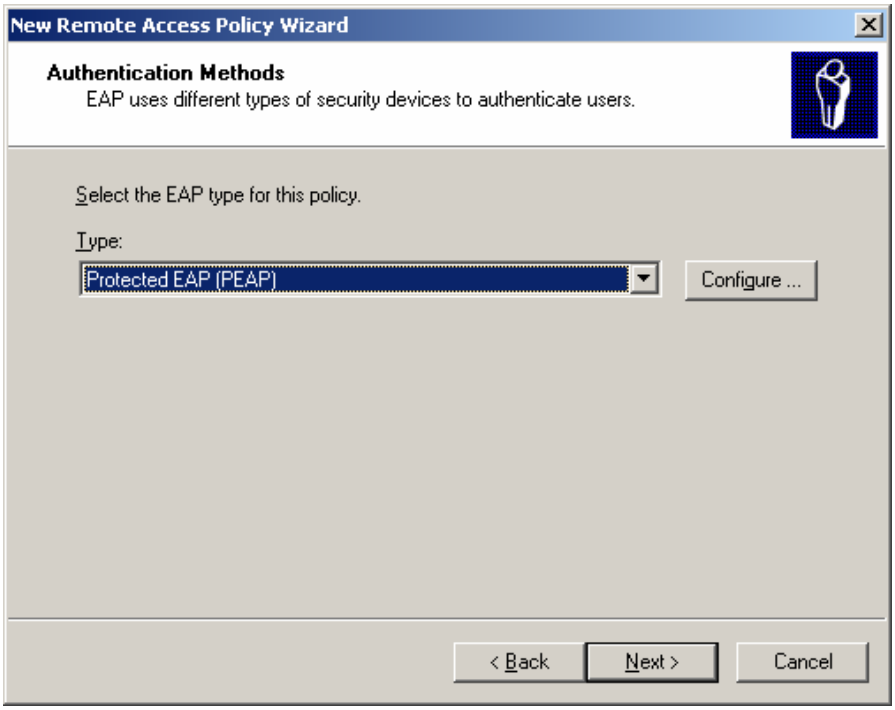
Figure III



Group access defined

Choose "Protected EAP (PEAP)" authentication. Then hit "Configure". (**Figure JJJ**)

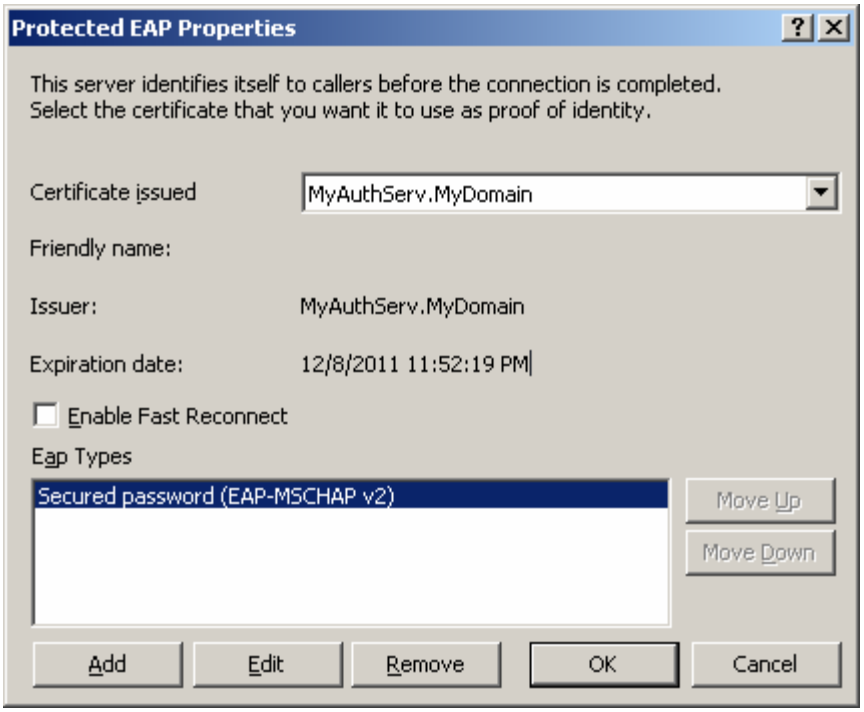
Figure JJJ



Authentication

Before you get to this page, you must either have a valid Machine Certificate from a Certificate Authority or you have already self-signed on yourself. Leave the rest of the settings like you see in **Figure KKK** and click OK.

Figure KKK



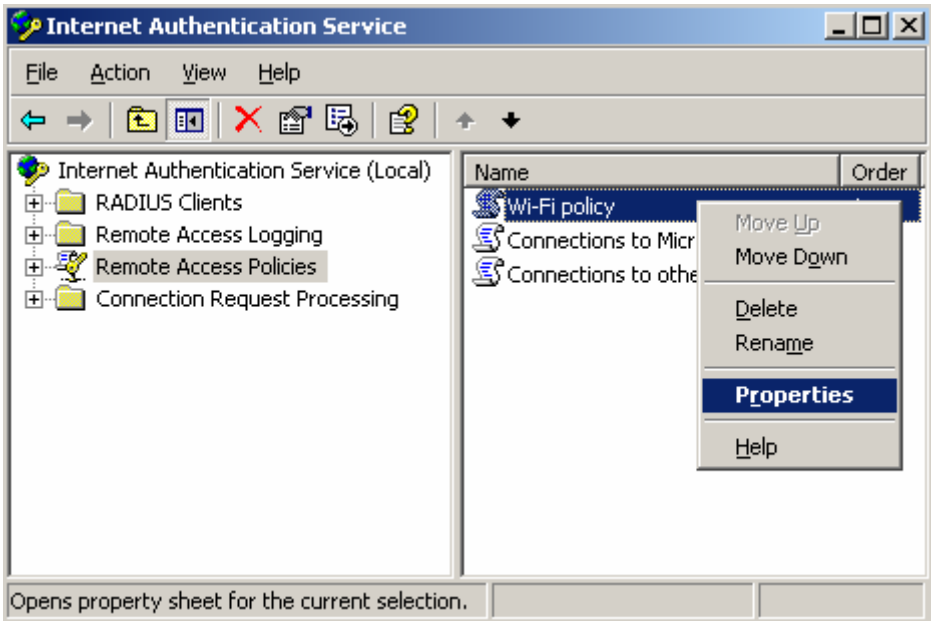
PEAP Properties

Finalize the remaining dialog box and you're finished making a new wireless authentication profile. Now we'll move on to fine tuning the configuration.

Tweak remote access policies

Once you complete the previous steps, you'll see a new Remote Access Policy called whatever name you gave it. The two default policies you see in **Figure LLL** the one we created are just there as samples and are disabled by default. We'll right click on it and hit "Properties".

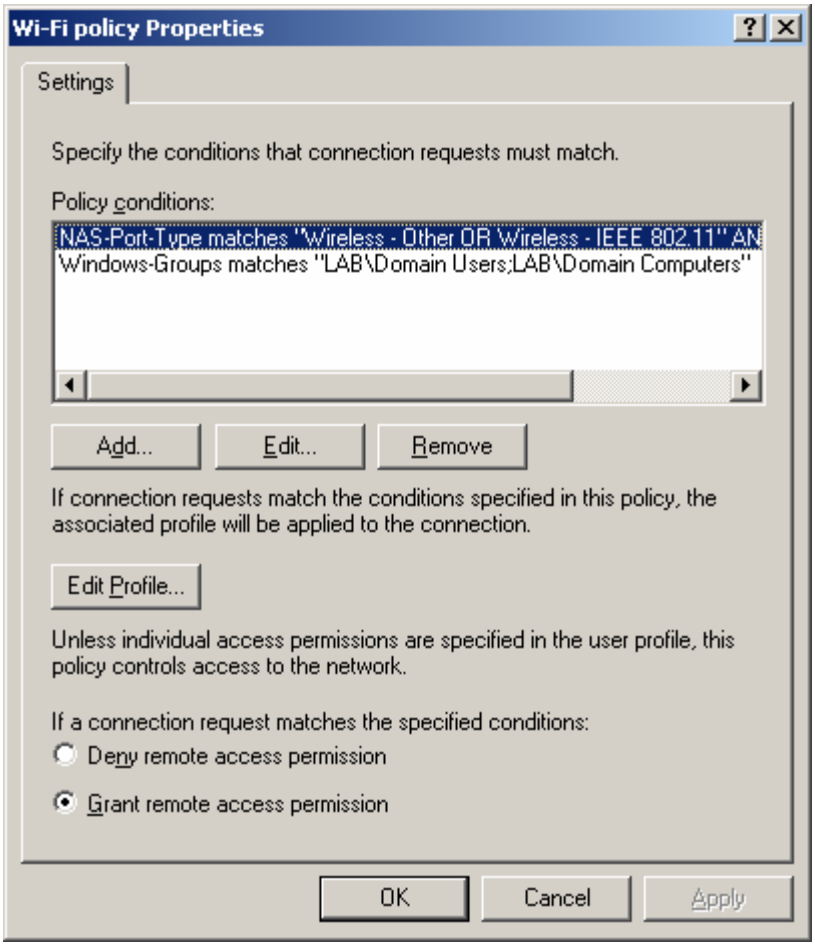
Figure LLL



Remote Access Properties

You'll notice there are two "Policy conditions" shown in **Figure MMM**. Note that there is an AND operator operating between the two conditions. That means both conditions must be true or else the policy spits out a rejection and it moves on to the next "Remote Access Policy". The first policy forces "Wi-Fi Policy" to only permit users coming in from 802.11 connections. The second policy is the permissible user or computer groups we set earlier. Click on "Edit Profile" to continue.

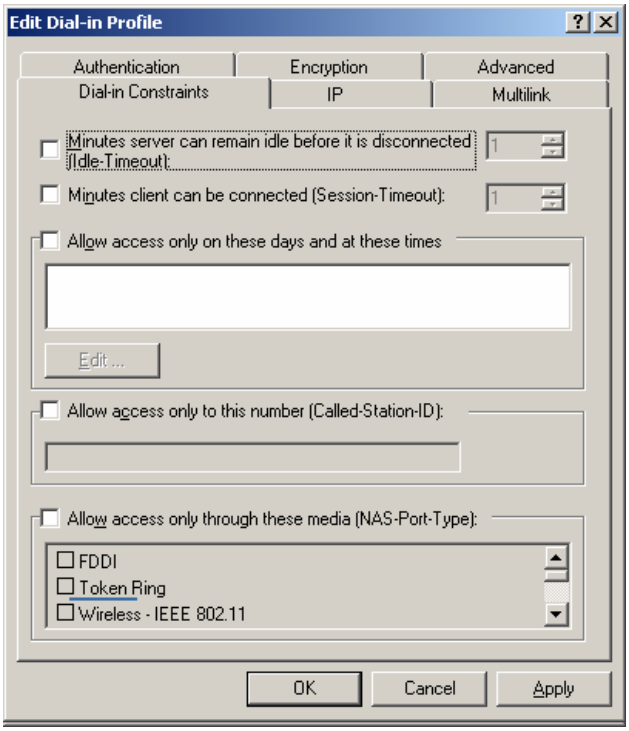
Figure MMM



WiFi Policy Properties

The "Dial-in Constraints" tab lets you set the dial-in and session limit restrictions (**Figure NNN**). It also lets you restrict the times people are allowed to log in.

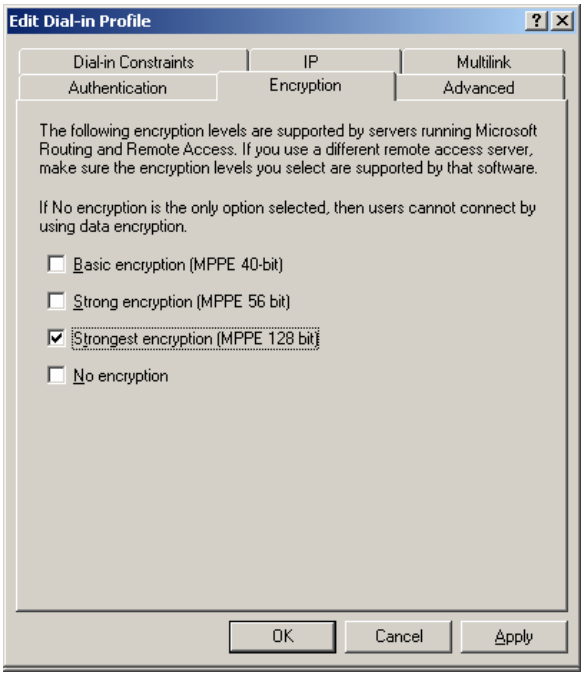
Figure NNN



Dial-in Profile

The "Encryption" tab is important for security (**Figure OOO**). You must uncheck the three insecure checkmarks to enforce maximum strength encryption.

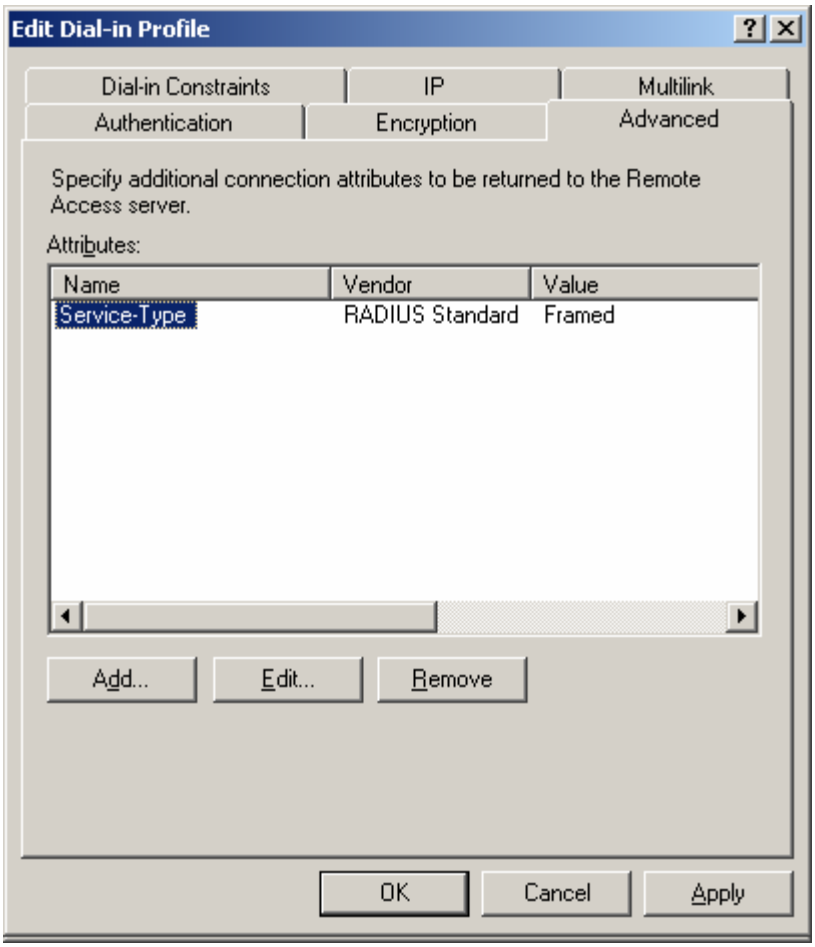
Figure OOO



Encryption

The "Advanced" tab (**Figure PPP**) is something we won't go in to now, but note that this is a very powerful tab for advanced features. With special RADIUS attributes configured on this page, you can do things like tell your Cisco VPN concentrator what user group a user belongs to so that the concentrator will set VLAN and firewall policies on that user matching their group rights. You can also do things like set VLANs or group association for an Aruba wireless switch which has a built-in firewall. We'll leave the details for a future advanced RADIUS configuration article.

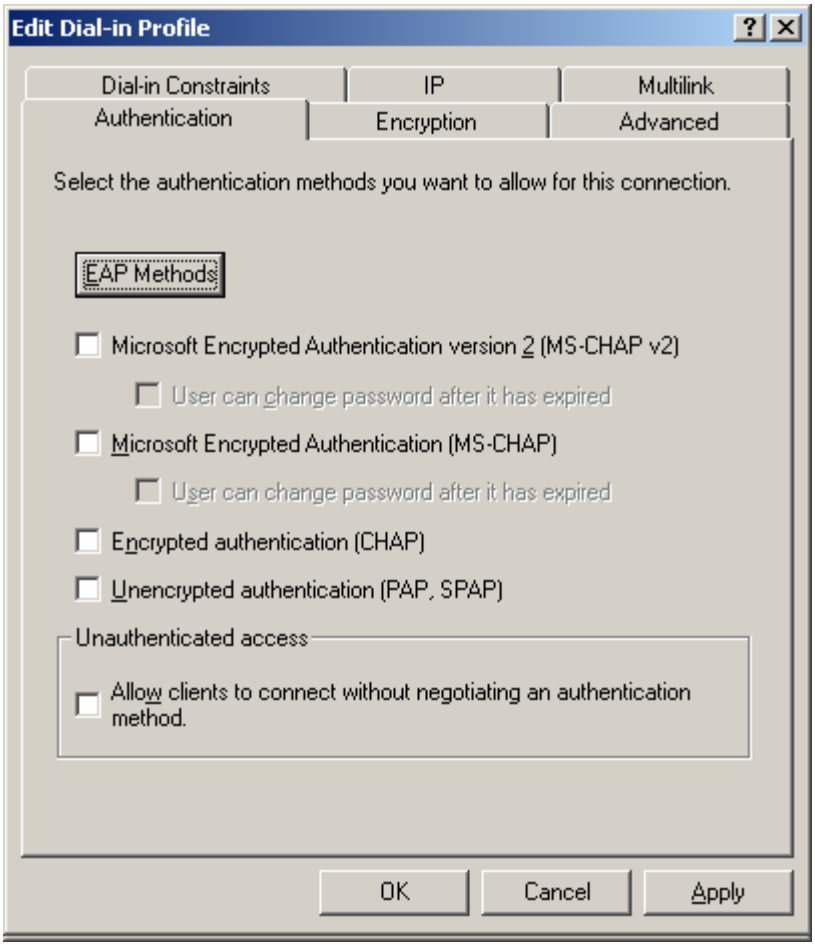
Figure PPP



Advanced tab

Under the "Authentication" tab, you can tweak the EAP methods (**Figure QQQ**). For wireless LAN PEAP authentication, you actually leave all the checkmarks alone. These settings are for more traditional RADIUS applications like a modem dialup service provider that proxies to your RADIUS server. Let's click on the "EAP Methods" button to see what it has.

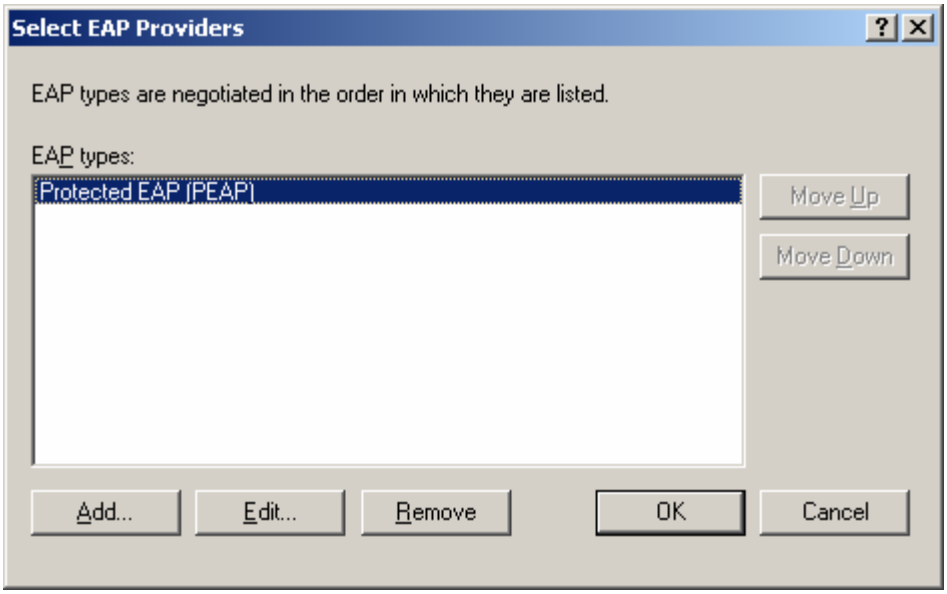
Figure QQQ



Authentication

Here you can edit the PEAP configuration. (**Figure RRR**) We already set these settings during the initial policy wizard. Click "OK".

Figure RRR



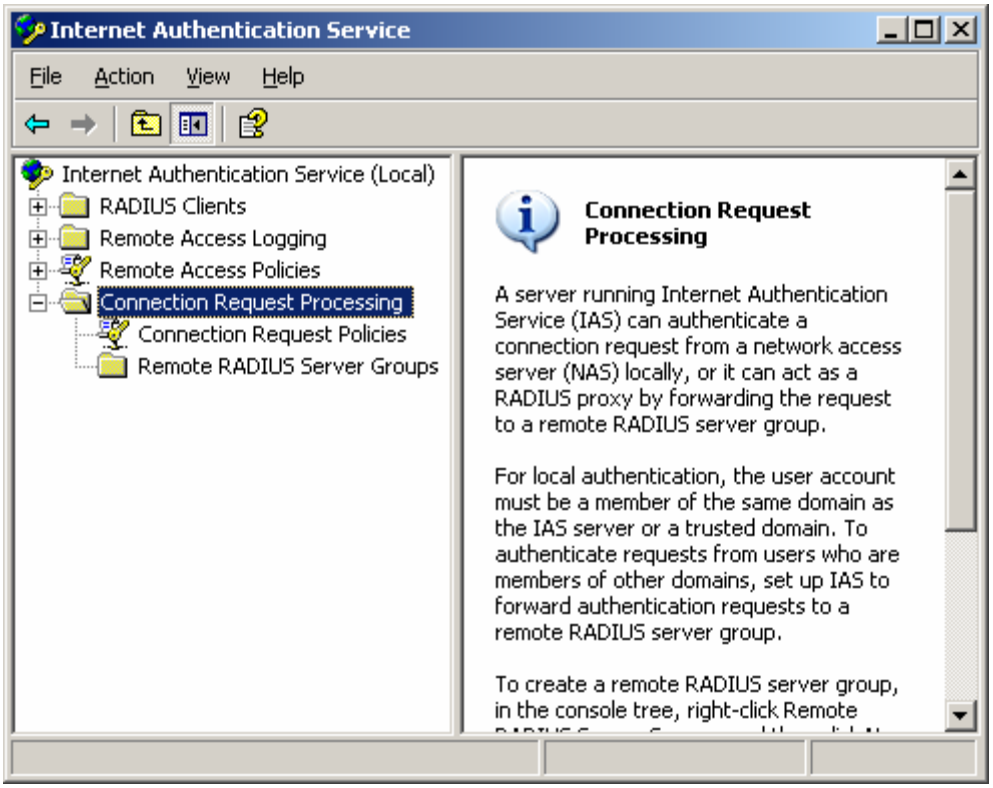
EAP Providers

You'll need to click OK one more time to get out of the Dial-in profile window.

This final section in the IAS interface is something we won't do in this article. (**Figure SSS**) I just wanted to give you a preview of what it does. The "Connection Request Processing" section lets you set advanced RADIUS relaying features. You have granular control of what kind of RADIUS requests you want to relay off to a different RADIUS server and which RADIUS requests you want to handle in the local "Remote Access Policy" engine.

You can even configure groups of RADIUS servers that you want to forward to. This allows IAS to participate in a multi-tier RADIUS environment. For example, if you have a user that isn't in your domain belonging to a business partner's network that needs guest access to your environment, you can forward to the RADIUS request to your business partner for them to process. There are even Universities that honor each other's students and staff by allowing a student to securely log in to any campus participating in the network.

Figure SSS



Connection Request

Backup and restore IAS policy

Finally, after all this work we want to be able to backup our RADIUS configuration and maybe even restore it on to a redundant RADIUS server. Microsoft gives you a simple command line tool for exporting and importing the RADIUS configuration.

To perform the backup operation, simply run the following command.

```
netsh aaaa show config c:\IAS.txt
```

Note that you can use any name for the file. You can use that file locally if you ever screw up the IAS configuration and you want to rapidly recover or if you want to copy the IAS setting to another IAS RADIUS server. To restore the IAS settings from the text file you created, simply run the following command assuming the correct path and file name.

```
netsh exec c:\IAS.txt
```

This makes it easy to rapidly deploy multiple redundant IAS RADIUS servers and it also gives you the peace of mind to rapidly repair an IAS server.

Configure Aironet access points for enterprise security

The Cisco Aironet class of wireless access points is very common in business and enterprise grade wireless networks. This tutorial is part of the ten-part Ultimate guide to enterprise wireless LAN security series and will tie in to the infrastructure described in the other nine articles.

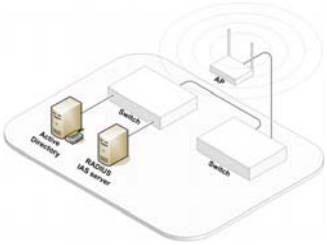
Enterprise class Wireless LANs with Aironet access points

In this tutorial, I will show you how to configure a Cisco Aironet IOS-based access point to setup the following things:

- Multiple Wireless LANs
- One VLAN (Virtual LAN) per virtual Wireless LAN
- Secure internal Wireless LAN that ties in to RADIUS and Active Directory
- Guest Wireless LAN with Internet only access

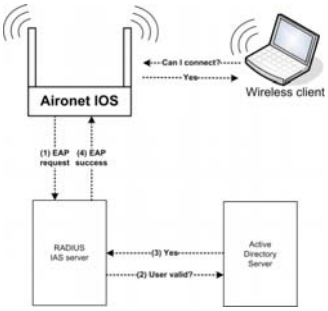
Figure TTT below shows a physical layout of the configuration while **Figure UUU** shows the logical link.

Figure TTT



Physical layout

Figure UUU



Logical link

Initial hardware setup

After you've removed the Aironet access point from the box and plugged in the power adapter, plug the supplied console cable to a valid serial port on your computer. If you have a laptop that doesn't have a serial port, you will need to get a USB to Serial adapter.

Once you boot up the Aironet access point, it will ask you to log in. The default user name and password are both usually set to *Cisco* by default. For example, here is a hardware installation guide for the Cisco 1100 series access point. Procedures for the Aironet 1100, 1200, and 1300 IOS-based stand-alone access points are all very similar. You will need to make sure you're running a more recent Aironet IOS for this guide to work since there are minor differences in the configuration and some features like multiple SSID broadcast weren't available in the older firmware.

Wiping the default configuration

The first thing I do with all the newer Cisco access points is wipe the default configuration on them. Older firmware didn't have any username and passwords assigned to them but the newer devices are different. Once you've logged in you'll need to type the following commands.

- enable
- write erase
- reload (confirm reboot)

Once the router is rebooted, you'll see a ">" prompt and you will be able to go in to "enable" mode without a password. You now need to enter global configuration mode by typing the old "*config t*" command.

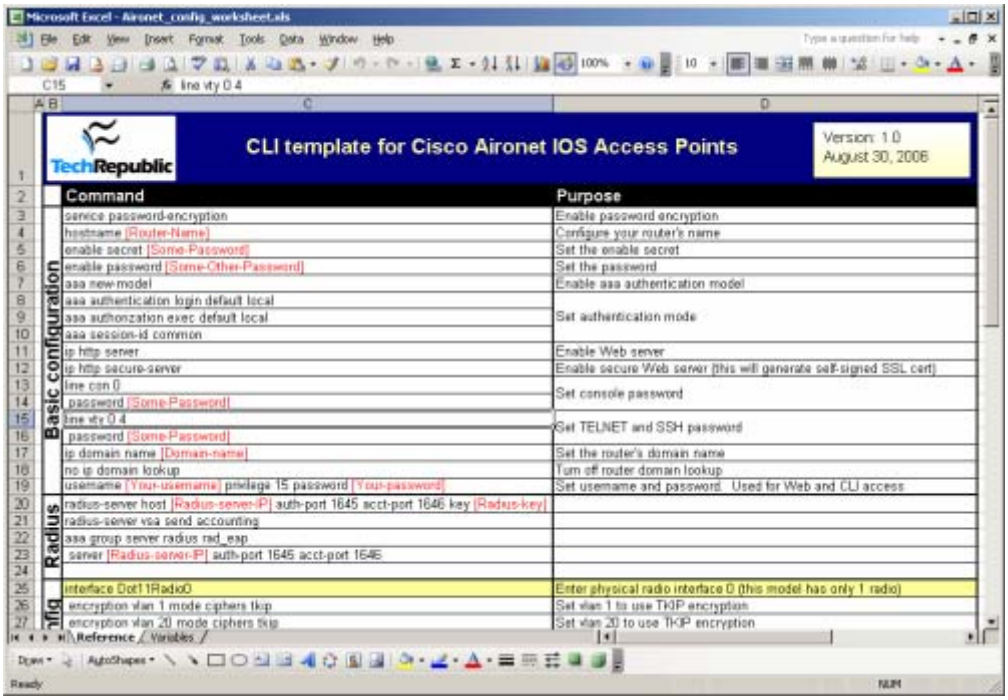
CLI configuration template for Aironet IOS

Since I've always thought that the Cisco configuration guides were too difficult to use with their inline comments and hints, I've created my own system of a configuration template in Microsoft Excel. Thanks to help from our development blogger Justin James, who wrote a quick replacement button that automatically generates a ready-to-use configuration output, we have a very useful tool for documenting and creating new CLI configuration files. For this specific tutorial, I've created this **Aironet IOS template** embedded with Justin's automation script.

How to use CLI template

Once you've downloaded the template for this tutorial, it's quick and easy to generate your own Cisco Aironet IOS configuration. All you need to do is fill out the yellow **section** shown in **Figure VVV** on the "Variables" tab page. The "Reference" sheet below in **Figure VVV** is the configuration template. It shows the configuration template with substitute variable names in RED colored fonts that are enclosed in [brackets].

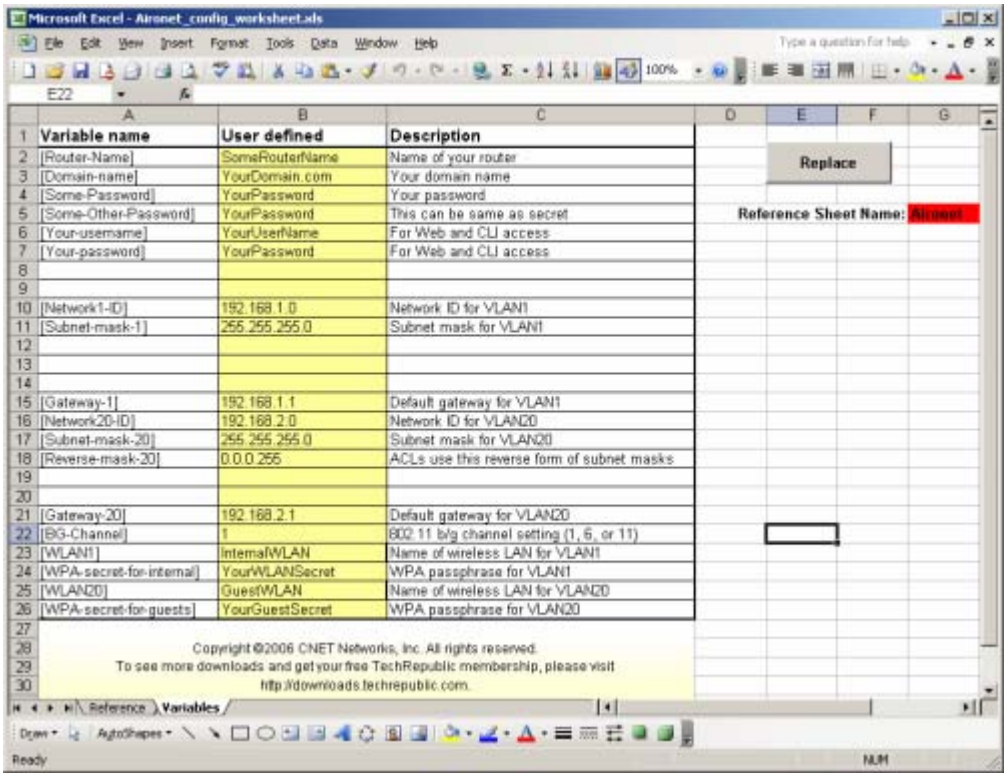
Figure VVV:



Configuration template

In **Figure WWW** below, the "Replace" button coded by Justin James will copy the content of the reference tab on to a new tab with the name Aironet (You can rename cell G5). You can use it multiple times and it will auto-increment the sheet names for each new configuration you create. This allows you to make slight modifications to the user defined variables to create a new sheet.

Figure WWW:



	A	B	C
1	**Variable name**	**User defined**	**Description**
2	[Router-Name]	SomeRouterName	Name of your router
3	[Domain-name]	YourDomain.com	Your domain name
4	[Some-Password]	YourPassword	Your password
5	[Some-Other-Password]	YourPassword	This can be same as secret
6	[Your-username]	YourUserName	For Web and CLI access
7	[Your-password]	YourPassword	For Web and CLI access
8			
9			
10	[Network1-ID]	192.168.1.0	Network ID for VLAN1
11	[Subnet-mask-1]	255.255.255.0	Subnet mask for VLAN1
12			
13			
14			
15	[Gateway-1]	192.168.1.1	Default gateway for VLAN1
16	[Network20-ID]	192.168.2.0	Network ID for VLAN20
17	[Subnet-mask-20]	255.255.255.0	Subnet mask for VLAN20
18	[Reverse-mask-20]	0.0.0.255	ACLs use this reverse form of subnet masks
19			
20			
21	[Gateway-20]	192.168.2.1	Default gateway for VLAN20
22	[BG-Channel]	1	802.11 b/g channel setting (1, 6, or 11)
23	[WLAN1]	internalWLAN	Name of wireless LAN for VLAN1
24	[WPA-secret-for-internal]	YourWLANSecret	WPA passphrase for VLAN1
25	[WLAN20]	GuestWLAN	Name of wireless LAN for VLAN20
26	[WPA-secret-for-guests]	YourGuestSecret	WPA passphrase for VLAN20

Reference Variables

Insert configuration on the Aironet access points

Once the configuration with your variables are created in a new worksheet, you literally copy the "Command" column with your customized settings (starting below the "Command" label) and paste it in to your console. Note that all the Excel formatting will be excluded from the paste command which is exactly what we want.

Also note that some commands take longer than others to insert because the device has to catch up so I would recommend you paste in a small section at a time and verify each of the commands executed properly without errors (some warnings notices are ok). The console is also known to drop certain statements at times if you paste too fast so make sure the router takes every single command. You can verify with the "show run" command to check the configuration. When you're satisfied, be sure to issue the "*write mem*" command to commit all the changes permanently so that the settings will remain intact the next time you reboot the router.

On the reference page, I've taken the time to label all of the commands with their purpose. This is for reference, learning, and documentation purposes. It would be wise to look through the entire reference page and understand what most or all the lines are doing. The more you understand the reference page the better off you will be in the long run.

The final Excel file is not only helpful for the configuration setup; it's also great for permanent documentation. The table format, the highlighting, and all the text formatting help make Cisco CLI more readable and understandable. You can also change the reference page to your liking if you want to modify the template to suit your own purposes.

Additional resources

- TechRepublic's Downloads RSS Feed XML
- Sign up for TechRepublic's Downloads Weekly Update newsletter
- Sign up for TechRepublic's Wireless NetNote newsletter
- Check out all of TechRepublic's free newsletters

Version history

Version: 1.0
Published: January 10, 2007

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to drop us a line and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team