Windows Security II
Active Directory Basics

James Madison University Dept. of Computer Science
March 23, 2015

1 Introduction

Working with a Windows machine on an individual level presents its own challenges. However, in larger corporations, additional technologies are used to help manage the company's computer assets. One of these main technologies is Windows Active Directory. This tutorial will teach you the basics about Active Directory. It will start with basic AD concepts and terms, then move on to how to actually find information in Active Directory, and end with some examples of typical administrative tasks that you would need to perform on an AD server.

2 Active Directory Basics

At its core, Active Directory is a structured information store, typically used on large-scale, corporate networks for tasks such as domain management, user authentication, and file sharing with access control. These tasks are vital to the operation of a flexible corporate network, where different users will need to be able to access shared files, log in with the same credentials at different desks if they travel, and try to collaborate with their coworkers in a secure manner. This section will discuss how Active Directory accomplishes these tasks, as well as how an Active Directory is structured on a high level.

2.1 Active Directory Components

In this tutorial we will be using a Windows 2008 server. As of the time of writing, that is the typical Windows server deployed to the Cyber Defense Competition team. In Windows 2008, the functionality of Active Directory greatly expanded past its previous purposes in Windows Server 2000 or 2003. [2] It is now structured into five main components, each encapsulating specific tasks to be accomplished. Below is a screenshot showing each of the five components in a Windows 2008 server configuration. A description of each of those components begins after the screenshot.

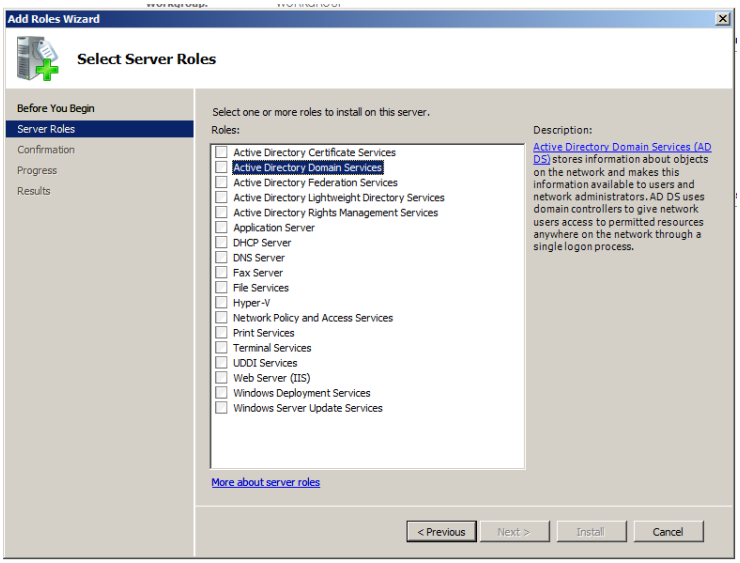

Figure 1: Different AD Options in Windows 2008 Server Configuration

2.1.1 Active Directory Lightweight Directory Services

AS LDS is a smaller version of Active Directory Domain Services which is to be used when an organization does not need all of the features of AD DS, but that still wants to use a directory service. Unlike AD DS, in AD LDS, Kerberos authentication, Forests and Domains, DNS dependence, replication across the network, and group policies are all removed. None of these features of AD DS are included in AD LDS. Instead, AD LDS is focused on performing the roles of phone book, consolidation store, and web-enabled authentication service.

2.1.2 Active Directory Federation Services

Federation Services (AD FS) provides Single Sign-On (SSO) service, which allows users to use one username and password to authenticate to multiple IT environments and make utilizing shared resources easier. This tool was originally released as part of the R2 release of Windows Server 2003, called simply "Federation Services," and has now been rolled under the Active Directory umbrella.

2.1.3 Active Directory Certificate Services

This component provides certification authorities for a network. It provides public-key certificates and key pairs that can be used to authenticate users via smart cards, encrypt and decrypt data, and manage those certificates and keys so they can be renewed and revoked as necessary.

2.1.4 Active Directory Rights Management Services

Although Active Directory Domain Services can help control whether or not a user could access a file. However, it does not have any capability to control what that user does once it receives the document. Active Directory Rights Management Services, or AD RMS, allows for more fine-grained access control on documents that can allow or deny users rights to do things such as email sensitive documents to unauthorized users.

2.1.5 Active Directory Domain Services

Active Directory Domain Services, or AD DS, is the component of the new AD that corresponds to the old functionality of Active Directory in older versions of Windows. This is the most commonly deployed component of Active Directory, and is the component that this tutorial will focus on. Because this was previously the only component of Active Directory and is so commonly the part of AD that is deployed in industry, for the remainder of this tutorial, unless I specifically note otherwise, I will refer to this part, AD DS, simply as Active Directory or AD. If I want to refer to a different component of Active Directory, I will use its full name or full acronym.

2.2 Active Directory Terminology – Structure

This section will discuss the ways in which Active Directory provides structure to large-scale networks and introduce the terminology used within AD.

The first thing to note when discussing the structure of Active Directory is that AD encapsulates both a physical and a logical structure. The physical structure is centered around the physical components that make up a network – the hosts on the network, the routers that make up the networking part of the network, the network configuration, and the bandwidth of the lines that connect different parts of the network. The logical structure is purely conceptual, and is used to try to match business practices and procedures of an organization to the Active Directory's structure. This logical structure should mirror how employees actually do their work and how the network is typically administrated.

At the end of the day, a properly configured Active Directory will result in the physical configuration being ignored by the users, as the logical structure of Active Directory can manage those resources on the end-user's behalf. For example, if AD has been well-configured, a user only need to know the name of the printer to actually print to it. They do not need to know which print server corresponds to that printer, nor do they need to know which domain that specific print server is a part of. This all sounds great, and in practice, it can be very useful. Now, we will discuss the terminologies and structures by which AD achieves these goals.

2.2.1 Domains

The **Domain** is one of the structural building blocks of Active Directory. A domain is defined by Microsoft as both an administrative and security boundary. All users within a domain typically will function under the same security policy and user-account policy. In general, if users should have different account policies, they belong in a different domain. For example, a user in the Sales department may need to have different account and password settings than someone in the Human Resources department who has access to personally identifiable information on a majority of the company's employees and applicants. This distinction means that the example corporation should consider separate domains for Sales and Human Resources users.

Each domain is defined by a few main characteristics. First, a domain must have at least one **domain controller.** The domain controller is a server that performs authentication on each user that attempts to connect to the domain. Second, the domain's directory database is replicated between all of the domain controllers on the domain. This process allows information to stay up to date across all of the domain controllers in order to avoid race conditions for bad authentication to occur.

2.2.2 Trees and Forests

A **tree** is the next level of abstraction above a domain. It is a hierarchical grouping of domains, where the top node is the "root" domain. All of the members of the tree are connected to each other, and each connection can be a bidirectional, transitive trust relationship, or an explicit, or one-way, trust relationship. Having a bidirectional trust relationship means that users in each of the two trusted domains can access some of the resources in the other domain. On the other hand, an explicit trust relationship can specify that users from one domain can access resources from the second domain, but that users in the second domain cannot access resources in the first domain. Each of the domains in a tree shares the same **namespace**, which at its core, is a logically structured naming convention. This namespace plays an important role within the structuring of DNS entries on the domain.

One level higher, a **forest** is a logical grouping of trees joined together by trust relationships. Each tree in a forest must have its own distinct namespace, but the trees in the forest share the same schema and global catalog (which will be discussed later.)

2.2.3 Objects and Organizational Units

In AD, everything can be thought of as an **object**. Examples of objects include everything from printers and files to users and groups. Each object contains data and additional descriptive information called attributes. Within AD, objects are stored in containers named **organizational unit** An organizational unit, or OU, is simply a container within a domain, used to store similar objects. OUs can be nested hierarchically to add additional structure to the object storage. However,

a single OU must be completely contained within one domain, and cannot span multiple domains. Typically, it is good practice to base the OU structure around the business practices of the company, such as different campuses or subgroups of employees that belong in the same domain.

2.2.4 Domain Schemas

The above structures do not just happen on their own. Each object must be clearly defined. This happens within the AD **schema**. The schema contains definitions of all object classes, including the attributes that make up those objects. Typically, the schema is created when AD is first architected, and is generally changed by software that needs to use the AD instance, such as a Microsoft Exchange server. Exchange needs a specific set of objects and attributes to exist within AD for it to function properly, but these objects and attributes are created programmatically by the Exchange server installer. A starting schema is installed whenever you install Active Directory.

3 Gathering Information with Active Directory

Part of the installation of Active Directory involved Microsoft-developed tools for the management of different aspects of Active Directory. In general, presence of these tools on a Windows Server 2008 machine shows that at least at some point, the corresponding components of Active Directory were installed on the server, if not showing that they are active at the time. These tools are all located in the "Administrative Tools" section of the start menu, as shown below.

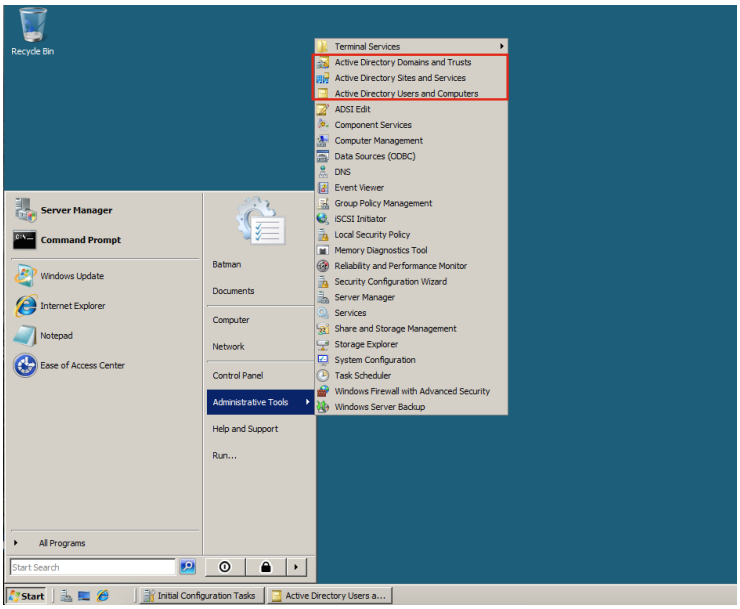

Figure 2: AD Tools in the Administrative Tools Menu

This tutorial will now walk through using some of these tools to explore what information can be stored in Active Directory and understand how to find and use it.

3.1 User Management

One of the main uses of Active Directory is storing information about different users on the network. This includes both user account information for the computers on the domain and "real-world" information such as phone numbers, addresses, and other personal information. The tool built-in to Windows Server 2008 for accessing and editing this information is the "Active Directory Users and Computers" tool, located in the administrative tools menu. Once you load the tool, you can navigate it very similarly to other Windows tools like regedit. The left-hand side has all of the main folders, which may have subdirectories as well. A screenshot showing one example of these is below.

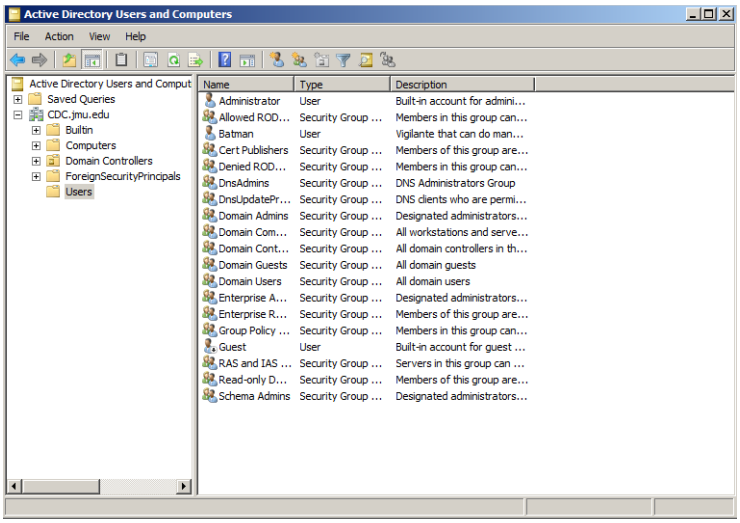

Figure 3: AD Users and Computer Tool

In the screenshot above, the "Users" pane was selected. This is where information on all of the users that Active Directory is aware of will be stored. If you double click on the "Batman" user, you can view the properties menu, which displays details about the Batman user. A screenshot of the "General" tab of this menu is below.

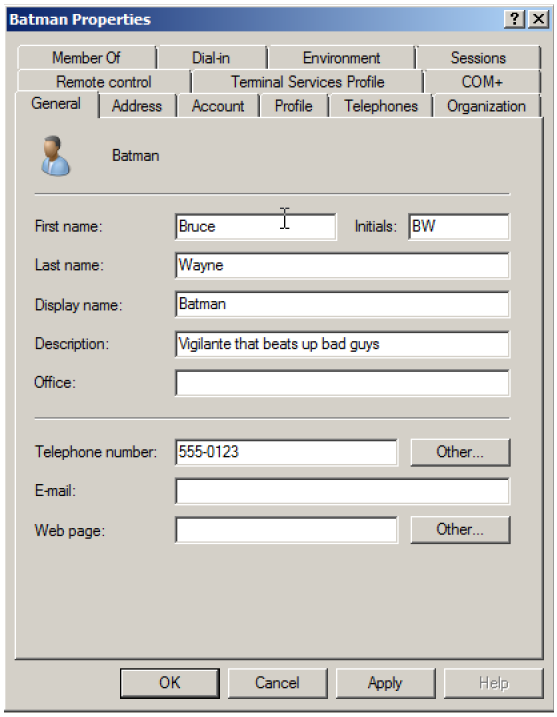

Figure 4: User Information in Active Directory

From within the properties menu, you can view (and change) contact information for the user, see which groups the user belongs to, and much more. In addition, this tool can also be used to look at the computers on the network, identify domain controllers, and more.

In a corporate environment, there are likely to be hundred or even thousands of users on a network. Looking through each of the users manually to find a phone number or even a specific user could be completely unfeasible in this type of setting. Furthermore, you are most likely to be using Active Directory when there are a large number of computers and users, otherwise, it would be overkill. Fortunately, Microsoft has provided a mechanism to search for users in a more efficient manner: queries. On the top-left of the Users and Computers tool, there is a "Saved Queries" field. In this case, there is one saved query already, which will search for any user whose name ends with "man". You can see this query if you expand the "Saved Queries" directory, as shown below.

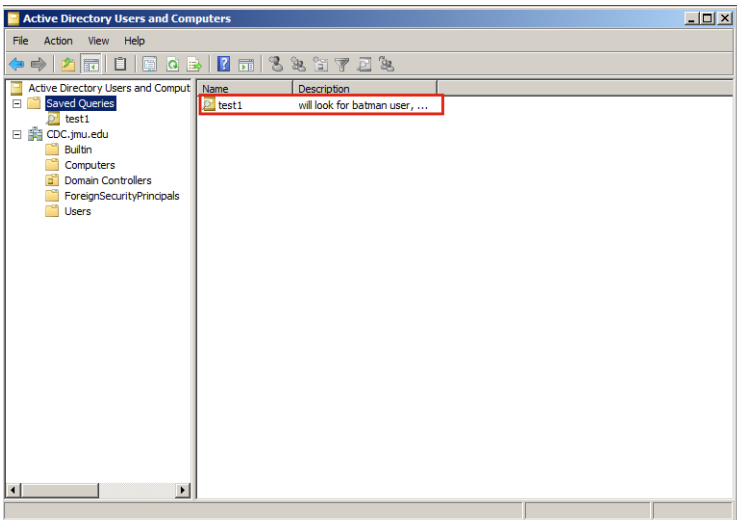

Figure 5: Viewing All Saved Queries

Next, if you left click on test1, it will actually show the results of running the query. In this case, there is only one user, Batman, that will be returned. However, were there to be another user named Aquaman, that user would appear in the query results as well. If you want to give it a try, right-click on the Users folder, click New User, and make the Aquaman user. Upon creating the user, come back to this query, right click the "test1" query and click the "Refresh" button, and you will see that Aquaman is now returned by the query as well.

If you want to see what the query definition itself looks like, or you want to change it, you can right click on the name of the query (in this case, "test1") and select "Edit..." This will open up a menu like the one shown below, and allows you to edit the name and description of the query, as well as the query definition itself, if you click the "Define Query" button. Likewise, if you do not want to edit this query but wish to simply create a new one, you can right click on the "Saved Queries" directory and select "New Query".

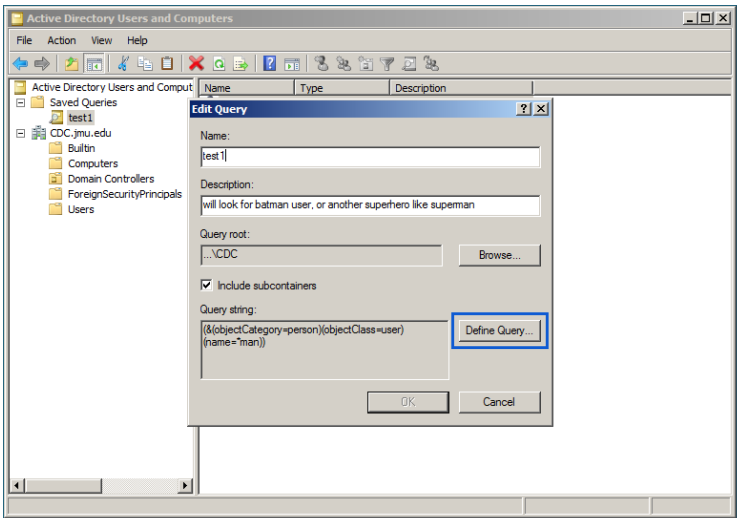

Figure 6: Editing a Query

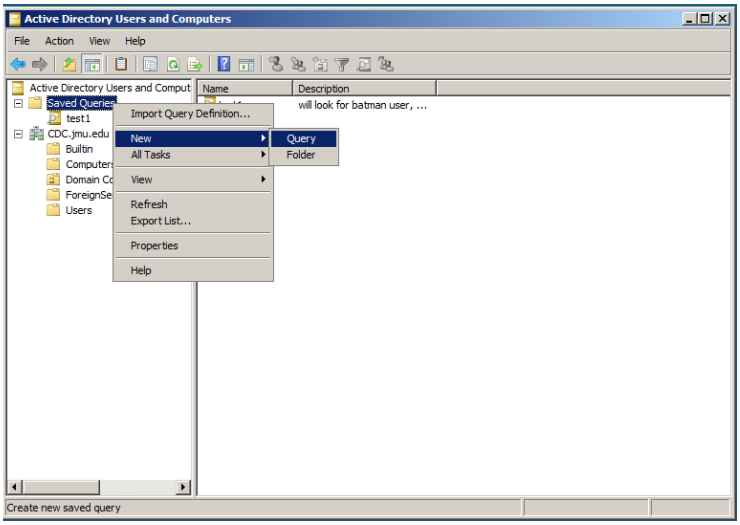

Figure 7: Adding a New Query

3.2 Site and Domain Management

There are two more tools included as part of the Active Directory Domain Services toolset: "Active Directory Sites and Services" and "Active Directory Domains and Trusts". Given that this tutorial is working on a virtual machine that is not actually administering a domain, those tools are not relevant to this

network. However, these work very similarly to the Users and Computers tool, as everything is stored in a hierarchical directory structure, and typically, right-clicking on an option or an object will give you a list of what you can actually do with them.

3.3 DNS

Because Active Directory is built on top of DNS, it is common for an Active Directory Server to also serve as a DNS server. This means that A records, which map the fully qualified domain names of machines on the domain to their IP addresses, are stored on the AD server. When administering a network, it can be important to understand exactly what your network topology looks like, and DNS helps shows a lot of information.

In order to view the A records for a given DNS zone, you must first open the DNS tool, located under the `Administrative Tools` menu, as shown below.

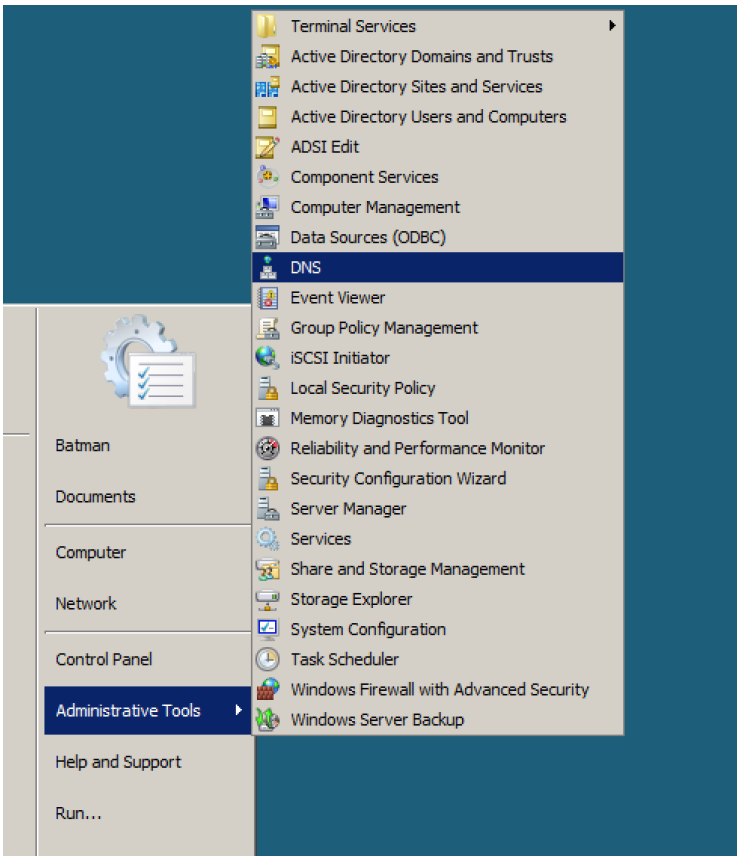

Figure 8: Opening the DNS Tool

Once you are in the tool, you want to click on the `Forward Lookup Zones` folder on the left, and then either expand the folder, or double click on the `CDC.jmu.edu` domain folder on the right-hand side of the tool. Once you have opened the actual domain folder, you can browse different information about the domain, including its A records, as shown in the screenshot below.

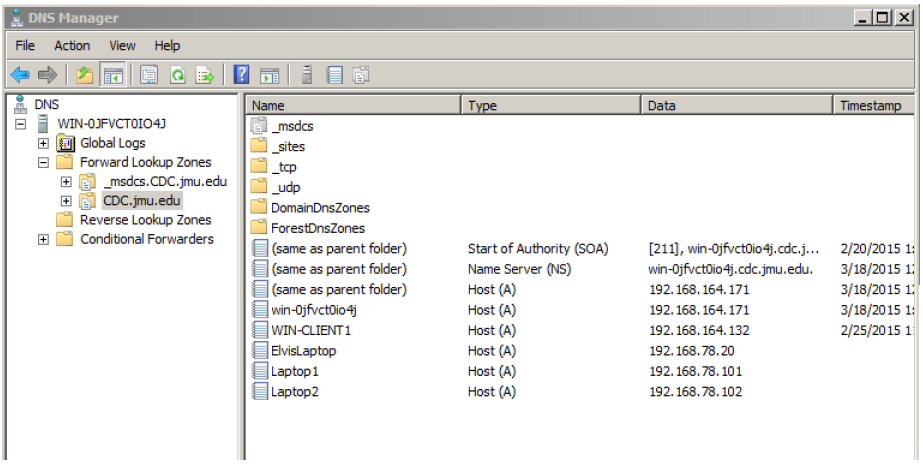

Figure 9: Viewing A Records on the CDC.jmu.edu domain

This shows that there are a few machines on the domain, including `ElvisLaptop`, which has the corresponding IP address of `192.168.78.20`.

4 Using a Securing Active Directory

Now that you know how to find various types of information on an Active Directory server, we will move on to work on securing the machine and performing typical administrative tasks on the AD server.

4.1 Using and Securing the Registry

One important part of understanding and securing an Active Directory machine takes place in the registry. The Windows Registry is a hierarchical data structure that is common to all Windows hosts, and is used to store some important machine configuration information. In this example, we will use a graphical tool to make a few changes to the registry that will increase the security posture of our AD machine.

4.1.1 Regedit - A Tool for Reading/Editing the Registry

One of the primary tools used for editing and reading values in the registry is `regedit`. This tool is built in to Windows and comes default with Windows

Server 2008 (and other versions of Windows as well.) You can run this tool by typing "regedit" into the run textbox in the start menu, as seen in the screenshot below.

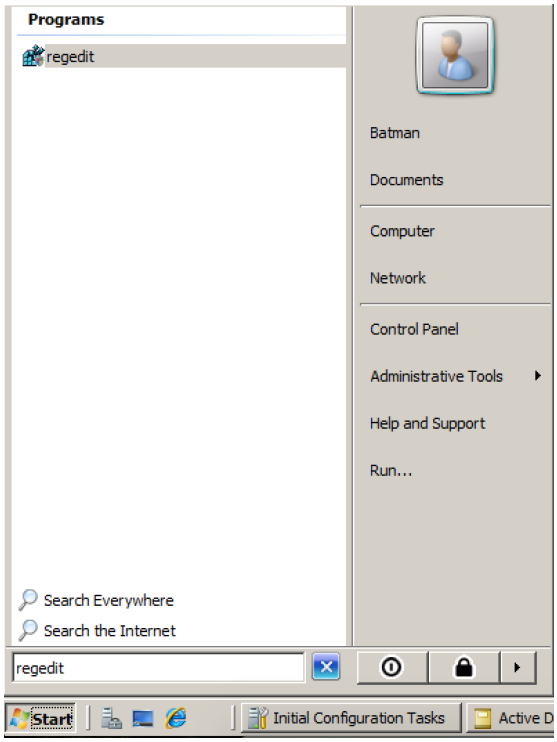

Figure 10: Launching Regedit

The registry editor allows you to browse the registry keys and subkeys, as well as view and set the values stored in each key. The screenshot below shows where each part of the registry can be viewed. The far left is where each key and subkey can be browsed in order to find the key that you are looking for. When you click on one of the keys, (`HKEY_CURRENT_USER\Environment` in the screenshot), the right-hand side of the screen shows all of the values and their corresponding data. If you double click on one of the values, a menu pops up that allows you to edit the value.

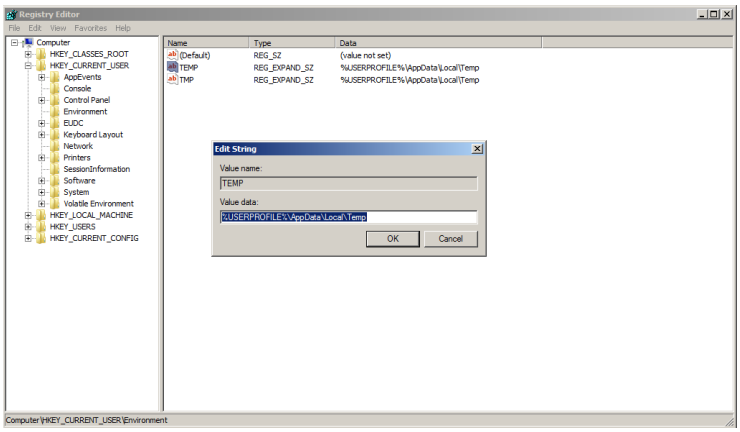

Figure 11: The Main Parts of Regedit

Instead of continuing to talk about the registry editor on a high level, we will now use the registry editor to view and modify some keys and values that are important from a security perspective.

4.1.2 Important Registry Values

When administering a system, one cannot assume that an attacker will never gain access to a legitimate user account on that system. In order to protect the machine against someone that has a user account, you need to try to protect important information on the machine so that even if an attacker gains access to the computer, it is harder for them to gain more privileges. One relatively easy way for an attacker to escalate privileges and gain persistence on the target host is to steal and crack the password hashes in order to get legitimate passwords on the system. This becomes even easier if the LM hash is enabled on the system, as it is an extremely weak hash. The LM hash is only used for legacy purposes, and is not needed in most current networks. Instead, Windows can use the NTLM hashing algorithm to create password hashes, which is much more secure than the LM hash. Fortunately, removing the LM hash is fairly straightforward when you can edit the registry. In order to disable the LM hash, you need to go to the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa` key and then set the value "NoLMHash" to 1. From here, you then need to change all of the passwords (which you can configure from within Active Directory by forcing each user to change their password on the next login), and the system will not generate LM hashes for these new passwords. If you do not change one of the passwords, its LM hash will still be stored.

Use the registry editor and make that change now. A screenshot showing the registry key with the value set to 1 is below.

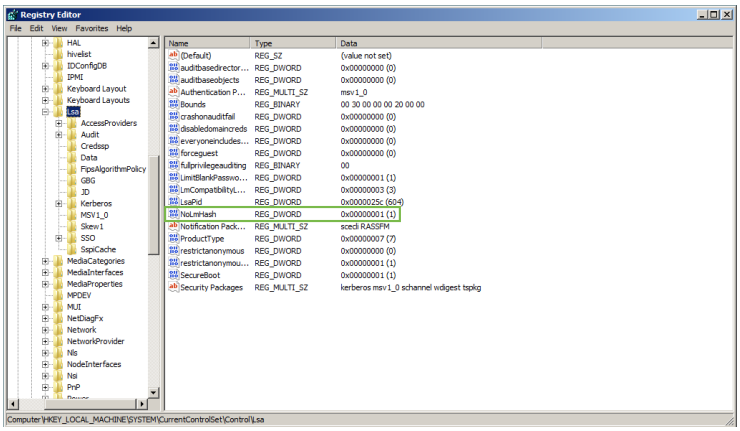

Figure 12: Turning Off LM Hashes in the Registry

There are two other keys that we will investigate in this tutorial are the Run and RunOnce keys. These keys are used to store pointers to programs that will be launched when a user logs in or when the system boots. This is one of two ways to get a program to run on login; the other being placing a shortcut to the program in the `C:\Documents and Settings\All Users\Start Menu\Programs\Startup` folder.

Typically, programs that need to add themselves to run on startup place themselves in the registry. If the program needs to run for all users on boot up, it will place itself in the

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` key.

On 64-bit systems, there is actually an entire subkey of the HKLM\SOFTWARE key, named "WOW6432Node" that is used for compatibility purposes. This is essentially a 32-bit SOFTWARE subkey that can be used by any 32-bit process. This means that 32-bit executables that need to be run on boot or login are placed in this subkey's run folder. This key's full path is:

`HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run`

If it only needs to be run for the current user, it will place itself in:

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` key.

There are two additional keys, both named RunOnce instead of Run (subkeys of the same `Current Version` keys) that will run a program once (at the appropriate time), and then delete its value out of the registry.

Navigate to the HKLM Run key on the server, and take note of what programs are in there. One of them, "VMWare User Process", seems to be legitimate. It is pointing to a file whose name and location is consistent with where the VMWare tools executable is typically stored, and given that this is a VMWare virtual machine, that makes sense. However, there is another value, named "Windows Server Process" that is in that key. If you look at it closer,

you can see that it points to a file named "tacos.exe" stored in the Batman user's Downloads folder. Because this is highly likely not to be a legitimate Windows Server Process, you should delete it from the registry. When you try to delete it, you will likely get a warning saying that deleting registry values can cause system instability (see below), but you can ignore that and delete it anyways.

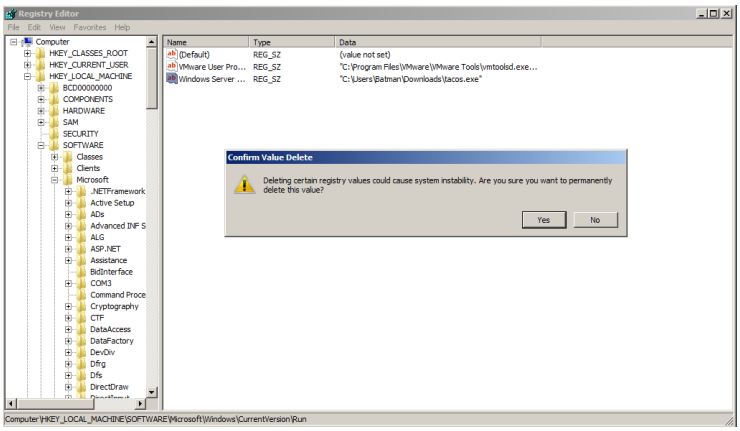

Figure 13: Warning When Deleting Registry Entries

4.2 Connecting Another Computer with AD

At this point, we have explored a lot about how Active Directory actually functions. However, we have not used another computer with the Active Directory server itself. This tutorial will walk you through how to add a Windows 7 machine to the domain and then log in using a domain account instead of using a local account. This will show that the computer is in fact on the domain and that Active Directory is actually doing authentication for that domain on the Windows 7 machine.

First, log onto the Windows 7 machine using a local account. This account name is "AdminUser", and the password is "badpassword". Next, you need to configure this machine to use the Active Directory Server as its DNS server. This is crucial, as Windows will use DNS to try to find the AD server, and if this is not done, it will not be able to find the domain. In order to do this, I referred to a Google tutorial [9] that walks through changing the DNS server for your interact. Go to the Control Panel, and open the "Network and Sharing Center". From there, click on "change adapter settings" on the top left (screenshot below.)

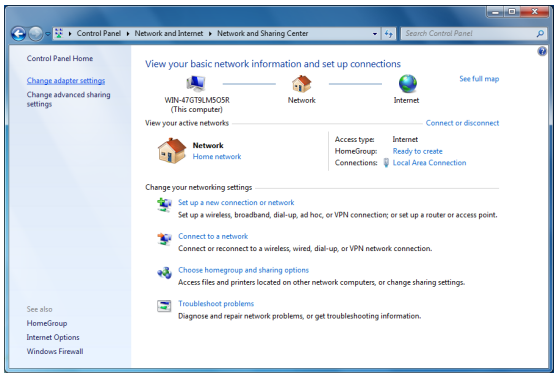

Figure 14: Network and Sharing Center

From here, select the "Local Area Connection" connection, double click on it, and go into the "Properties" menu.

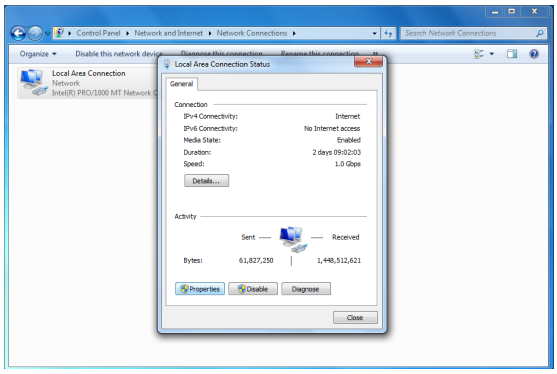

Figure 15: Configuring the Adapter

Next, select the IPv4 part of the adapter, and click on its Properties menu.

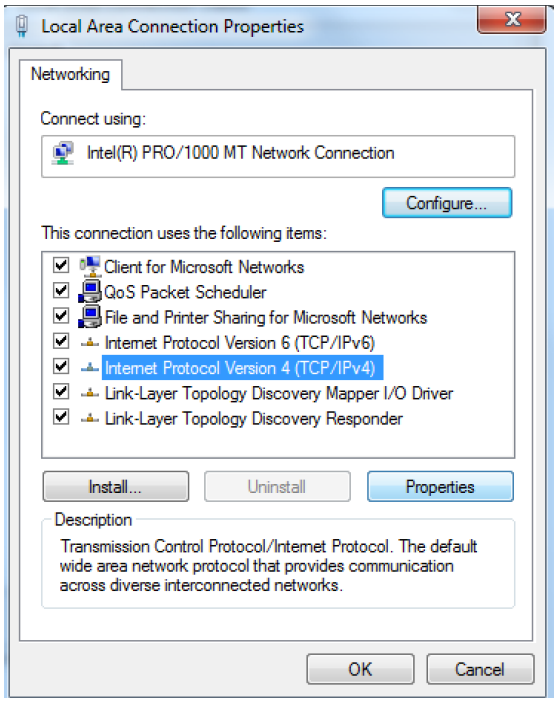

Figure 16: Configuring IPv4

Once you are in the properties menu, you can manually set the DNS server to use the AD server as its DNS server. Run `ipconfig` on the server to get its IP address, and put that value as the Primary DNS server for the Windows 7 machine, as seen below. In the example below, the IP of the AD server was `192.168.164.171`.

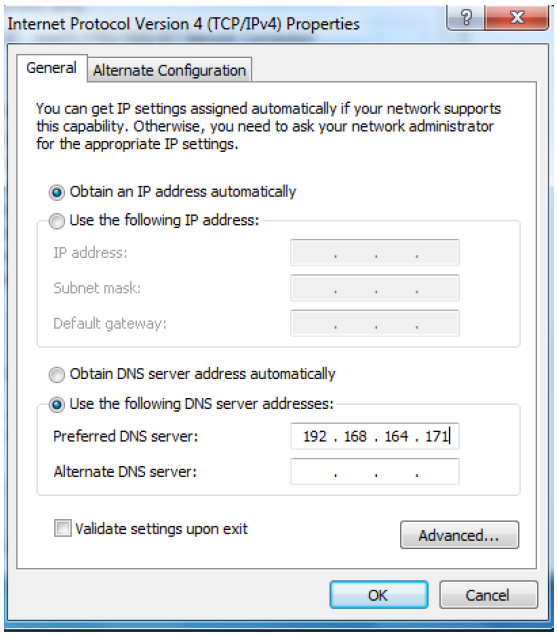

Figure 17: Set the AD Server as the Machine's DNS Server

Now, you only need to go back to the adapter menu, disable it, and re-enable it (by clicking on it in the adapter menu), and it will not use the AD server as its DNS server.

Before you can join the domain, you need to make sure to add the computer to the domain from the Active Directory server. In order to do this, go back to the AD server and open up the "Users and Computers" tool that you used earlier. From here, right click on the Computers directory on the left, and add a new computer. In the screenshot below, you can see that I added a computer named "WIN-CLIENT1". In the real world, you would have a specific schema for these names that determines what you name the machines. In this case, almost any name will do, as long as it starts with "WIN" for the Windows client.

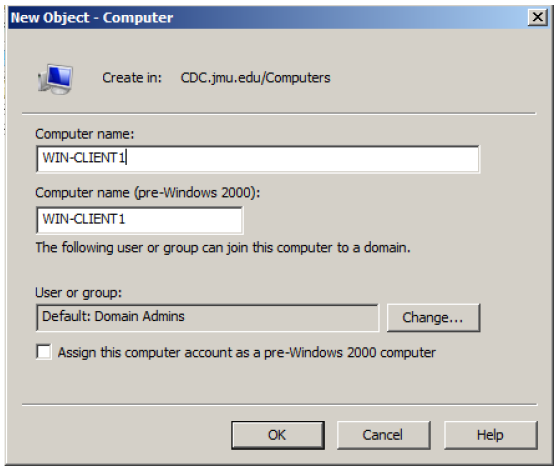

Figure 18: Adding a Computer to the Domain

Now that the computer has been added to the domain (which gives it permission to connect in to the domain), you can go back to the Windows 7 machine and actually join the domain. Click on the start menu, right click on "Computer", and click on the Properties menu (as shown below.)

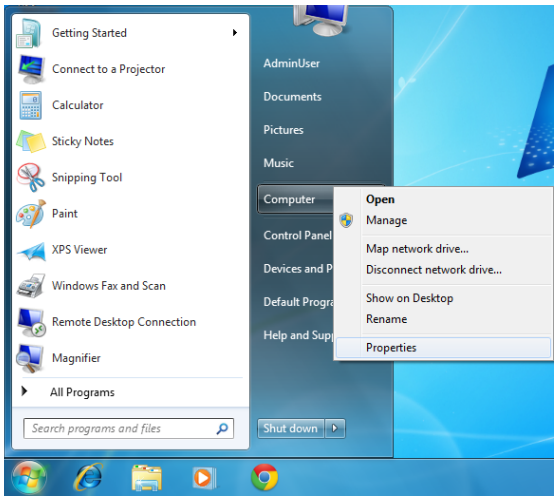

Figure 19: Opening the System Properties Menu

Once you've opened the properties menu, click on the "Change Settings" button, highlighted below.

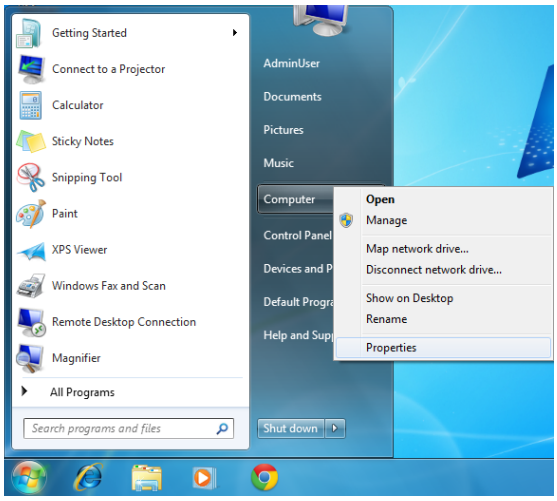

Figure 20: Opening the System Properties Menu

From here, click on the "change" button in the bottom right of the properties menu. This will open up a menu like the one below. From here, you need to enter the name of the computer (which you just used earlier) and the name of the domain, which is "CDC.JMU.EDU". enter those into this menu and click "ok". As long as you have entered the rest of the information properly, you will get another message, saying that you have joined the domain.

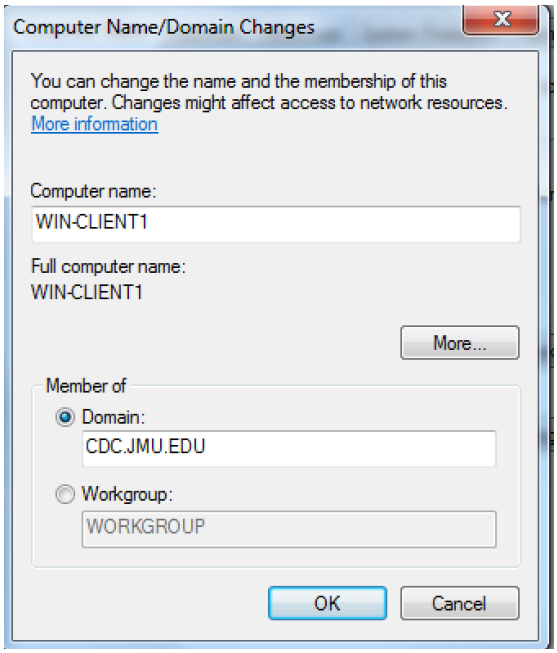

Figure 21: Joining the Domain

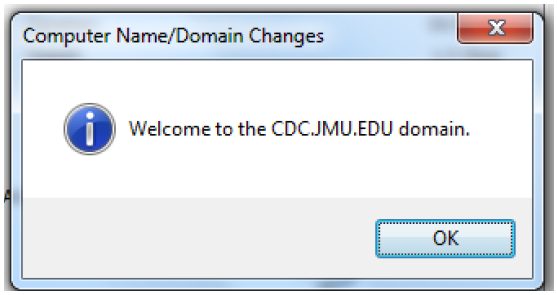

Figure 22: Domain Joined

In order for this change to actually take effect, you will need to reboot the Windows 7 machine. Do this now. Once the machine has rebooted, you will see that the regular users are now named "WIN-CLIENT1/[Username]", and you can click on "other user" on the main login screen. If you click on this "other user" button, you will see, as shown in the screenshot below, that the domain is now set to CDC. Now, try to login with a domain user: Batman, with the password for that domain user, "Cyb3rDefensePassw0rd". You will see that this successfully does log you in. This shows that the machine has been connected to the domain, and that AD is now managing domain user authentication.

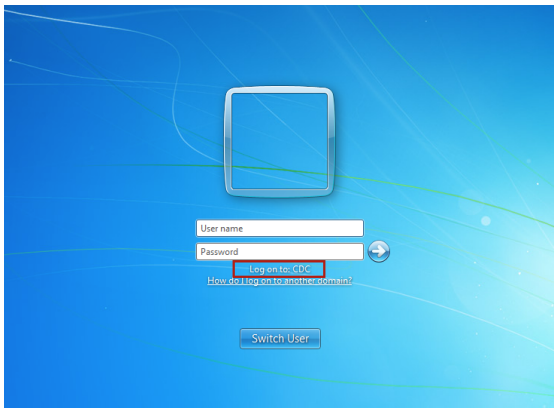

Figure 23: Logging onto a Domain User

4.3 Setting Up a DNS Zone

In a typical enterprise network, the Active Directory server serves as the DNS server as well. This is mainly done because AD is itself built on top of DNS, and Active Directory cannot function without DNS working on the network.

This means that when a the company needs a new DNS zone (for example, if they acquire another company or make a name change), you will need to create a new DNS zone. In this case, we will simulate an example where you need to add a new zone for the digital forensics group. We will walk through making a new domain, named `forensics.jmu.edu`, which will be administered by this AD server.

The first step in adding a new zone is to open up the DNS tool, which you used earlier in the tutorial to look up A record values. Once the tool is open, right click on the `Forward Lookup Zones` folder on the left, and click `New Zone...` as shown in the screenshot below.

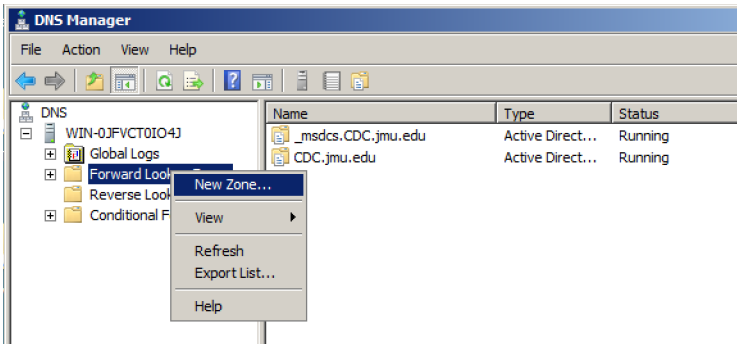

Figure 24: Launching the New Zone Wizard

This will open up the New Zone Wizard, which walks you through creating

a new zone. For the purposes of this tutorial, you will make a primary zone (the first option) and opt to store the Domain in Active Directory (at the bottom of the first menu). Go to the next page, and leave the option as it is: to replicate to all DNS servers in the CDC.jmu.edu domain. The next part of the wizard is where it asks you for the name of the new zone. This is where you enter `forensics.jmu.edu`. The next page will ask you how you want updates to occur. Select the first option, which is used by default. This is the more secure version that still allows dynamic updating over the network. Now, the only thing left is to click "Finish", and the new zone will exist, as seen in the screenshot below.

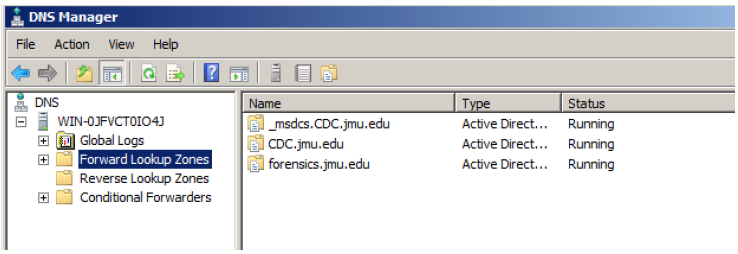

Figure 25: Forensics.jmu.edu Domain is Created

From here, you can add new A records by entering the domain folder (like you did earlier in the CDC domain), and then right clicking and selecting "New Host", as shown below. This will open up a screen where you can enter the machine's name and its corresponding IP address.

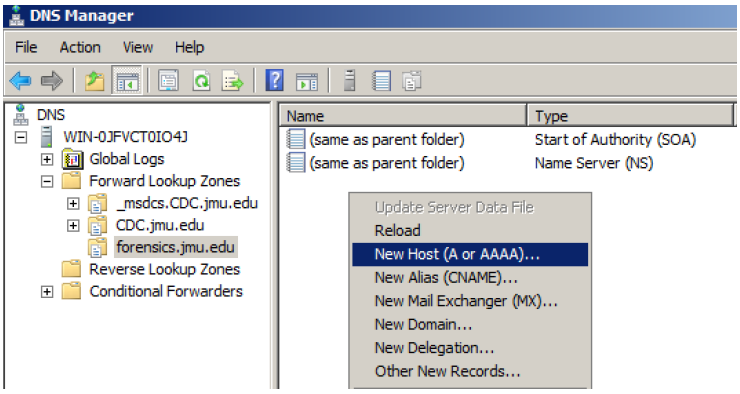

Figure 26: Adding an A Record to the New Domain

4.4 Creating a Share Drive

Another way that Active Directory services are used to increase productivity on a domain is through the use of Shared Folders. These folders are managed

by the Active Directory server, and can be accessed by the users that have permissions from anywhere on the domain. We will now set up one of these share drives so you can learn how to use them if needed.

In order to do this, you need to open up the `Computer Management` tool, which is one of the Administrative Tools. Once you have loaded the tool, you need to click on the Shared Folders/Shares folder on the left, as seen in the screenshot below.

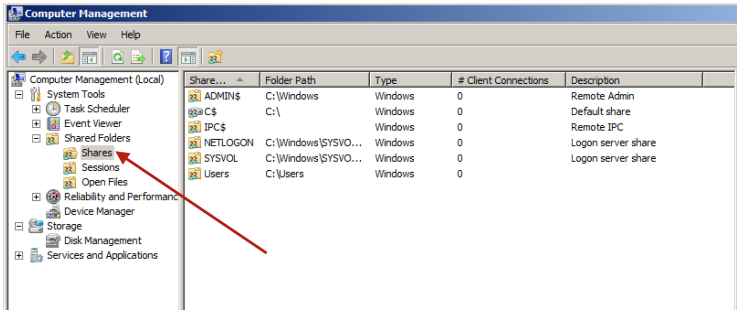

Figure 27: Viewing the Shares

From here, right click on the white space under the listed shares and select "New Share." This will open the "Create a Shared Folder Wizard", which is fairly straightforward. A screenshot of the first page of this wizard is below.

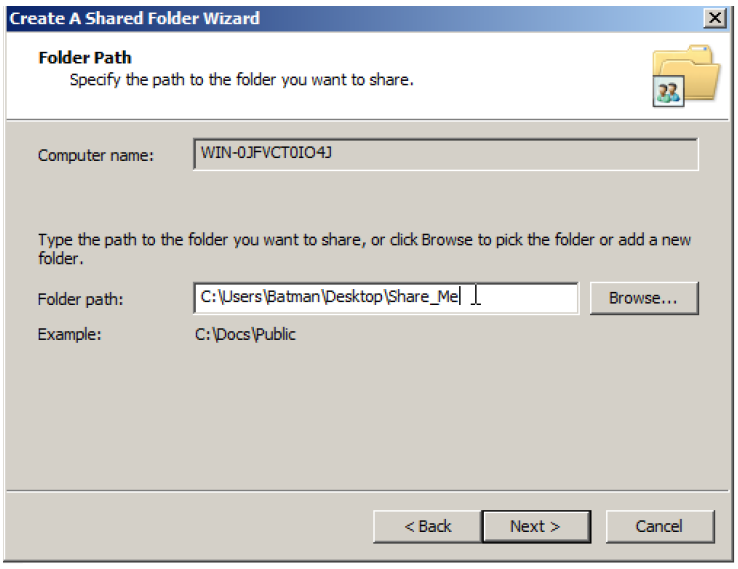

Figure 28: Starting the New Share Wizard

One thing to note is that during the creation of the share, you can determine

what permissions you want to grant users for the network share. This sets share permissions across the domain. If you want more fine-grained control, you either need to select "Custom" on the permissions part of the wizard (shown below), or go to the folder after it has been created and edit the permissions manually like you would any other share drive.

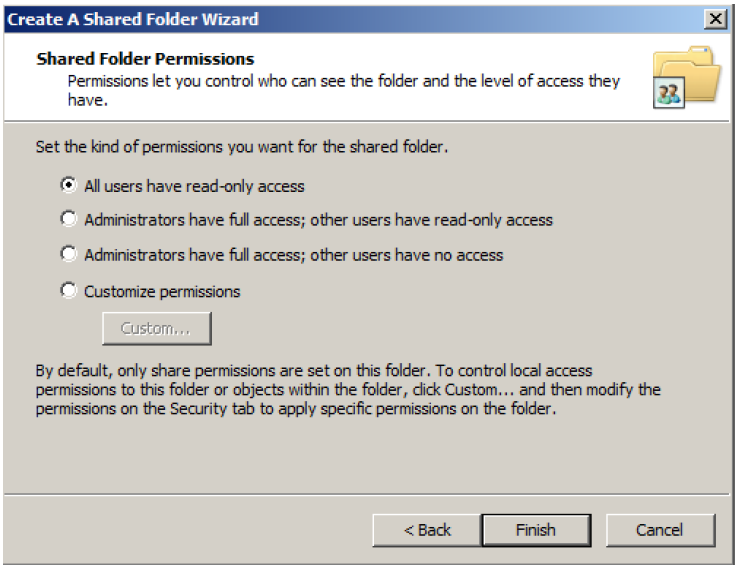

Figure 29: Setting Network Permissions

That is all that is needed to make a network share.

5 Example Active Directory Attack

On February 10, 2015, Microsoft released a security update to patch vulnerability MS15-011. As described in an article on Ars Technica [4], this vulnerability, which existed in Active Directory for at least 15 years, and still unpatched in Windows Server 2003, allows attackers that can monitor local network traffic to launch a man-in-the-middle attack on users on the sniffed network and execute arbitrary code on their machines. The picture below illustrates how one typical attack scenario unfolds, and is borrowed from a Microsoft technet blog post [5]. An attacker must first be able to sniff network traffic to find that the (soon to be) victim machine is attempting to download a specific file from the AD domain controller. The attacker can then perform ARP Poisoning on the victim to make sure that the next request for that file goes to his own machine instead of the remote server. When the victim asks for the file again, they will instead be given the malicious code. These files that are retrieved are typically .bat or other executable files, so the attacker could immediately gain remote code execution on the target machine.

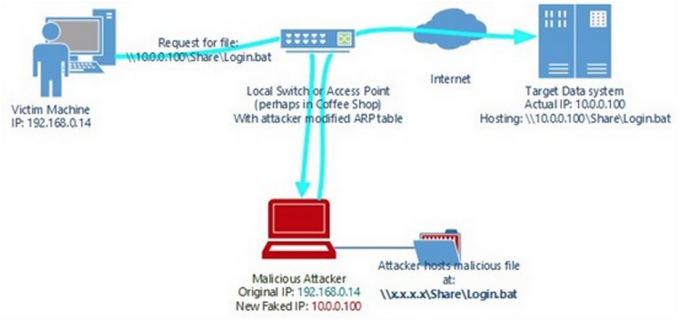

Figure 30: Attack Flow for MS15-011

This flaw existed because some Group Policies did not require SMB-signing, which would force the requesting machine to verify the sender of the .bat file that it is requesting. By default, the SMB client did not require signing, so it was relatively simple to perform this attack, especially because requests for files were often send over the network in an unencrypted format.

At the time of writing, this critical security flaw has been patched. As noted in the first Windows Security tutorial [6], patching is a critical part of defending any system, and in this case, is one of the only ways to protect against this type of attack against an Active Directory enabled network. For more information about how to install patches on a Windows machine, please refer back to that tutorial [6].

6 Conclusions

In this tutorial, you learned about the basics of how Active directory is structured, learned how to find information stored in Active Directory, and how to perform typical administrative tasks on an Active Directory server, like adding users, using the DNS tool to set up domains, and adding a Windows 7 machine to the domain itself.

This will allow you to have a better understanding of how Windows networks are administered and make you more prepared to work on those types of networks. Although these types of networks are not typically used in an academic setting, they are prevalent in corporate networks around the world, and understanding these technologies is vital to system defense in the real world.

7 Bibliography

[1] - https://msdn.microsoft.com/en-us/library/windows/desktop/ms724877%28v=vs.85%29.aspx

[2] - Active Directory for Dummies, by Steve Clines and Marcia Loughry

[3] - https://msdn.microsoft.com/en-us/library/aa746468%28v=vs.85%29.aspx

[4] - http://arstechnica.com/security/2015/02/15-year-old-bug-allows-malicious-code-execution-in-all-versions-of-windows/

[5] - http://blogs.technet.com/b/srd/archive/2015/02/10/ms15-011-amp-ms15-014-hardening-group-policy.aspx

[6] - https://users.cs.jmu.edu/tjadenbc/Bootcamp/2-WindowsSecurity.pdf

[7] - https://msdn.microsoft.com/en-us/library/windows/desktop/ms724878(v=vs.85).aspx

[8] - https://filebox.ece.vt.edu/ ece1574/spring14/devenvinstall.html

[9] - https://developers.google.com/speed/public-dns/docs/using#testing

8 Appendix A - Configuration Notes

Below are some notes about the configuration of the server that you may need. Most of these are not needed for completion of the tutorial, but I've noted them here just in case.

batman user password: Cyb3rDefensePassw0rd

admin password: AdminPassword

AD password: ADAdminPassword