U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE® PROGRAM

CyberForce® 101

Windows

October 2023

cyberforcecompetition@anl.gov

Intro to Windows

☰ **Pre-Requisites**

- **Intro to PowerShell**

Overview

Windows is an operating system designed by Microsoft. The most common editions of Windows for home computers are Windows Home and Windows Professional. There are also Windows Enterprise, Windows Education, and Windows Server editions. The features of a system depend on which edition you are using. Unlike Linux, Windows uses a GUI operating system on all editions of it.

In this guide, we will go over various aspects of Windows, including remote desktop, how to manage users and files, and Windows Active Directory.

Remote Desktop

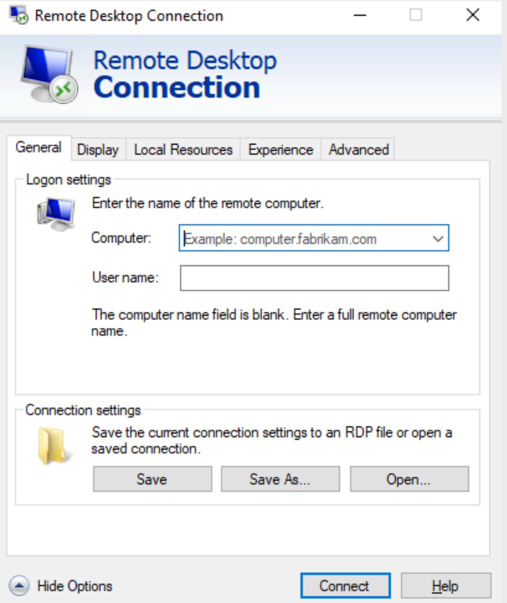
RDP, or Remote Desktop Protocol, is used for connecting to a desktop computer remotely. It is the most commonly used protocol and is available for most Windows operating systems.

Remote desktop users can access their desktop, open and edit files, and use applications like they are actually sitting at their desktop computer. Users are actually accessing their physical desktop computer and can only use files and applications that are saved locally on the desktop.

The RDP protocol opens a dedicated network channel for sending data back and forth between the connected machines, using port 3389. Mouse movements, keystrokes, the desktop display, and other data are sent over this channel using TCP/IP (the transport protocol used for most types of Internet traffic). RDP encrypts all data for security.

To set up the PC you want to connect to for remote connection, make sure to enable it in the settings. Select `Start > Settings > System > Remote Desktop` and turn on `Enable Remote Desktop`.

To use Remote Desktop to connect to the PC you just set up, search for `Remote Desktop Connection` with the search bar on the taskbar. In Remote Desktop Connection, type the hostname or the IP address of the computer you are wanting to connect to. You will also need to type the username of the account you'll use to log in.



Enter the name of the remote computer.

Computer: Example: computer.fabrikam.com

User name:

The computer name field is blank. Enter a full remote computer name.

Connection settings

Save the current connection settings to an RDP file or open a saved connection.

Remote Desktop Connection Screen

If you want to launch the RDP client through PowerShell or command line use the following command: `mstsc \v:10.10.10.10:3389`.

Managing Users and Groups

You can manage users and groups with the control panel or use PowerShell commands, which is much more efficient. You can use the built-in PowerShell module, `Microsoft.PowerShell.LocalAccounts` to manage local users and groups. This module lets you create or delete users and security groups and add or remove users from groups.

There are 15 cmdlets in the `LocalAccounts` module, and you can display the full list of module cmdlets with the command `Get-Command -Module Microsoft.PowerShell.LocalAccounts`.

- `Add-LocalGroupMember` – add a user to a local security group
- `Disable-LocalUser` – disable a local user account
- `Enable-LocalUser` – enable a local user account
- `Get-LocalGroup` – get information about a local group
- `Get-LocalGroupMember` – view the list of users in a local group
- `Get-LocalUser` – show information about a local user
- `New-LocalGroup` – create a new local group
- `New-LocalUser` – create a local user
- `Remove-LocalGroup` – delete a local group
- `Remove-LocalGroupMember` – remove a member from a local group
- `Remove-LocalUser` – delete a local user
- `Rename-LocalGroup` – rename a local group
- `Rename-LocalUser` – rename a user

- `Set-LocalGroup` – change group settings
- `Set-LocalUser` – change user settings

We will go over a couple of examples of these cmdlets at work in the text below. For more examples, view source 4.

If we want to create a new user, we use the `New-LocalUser` cmdlet like shown below.

```
New-LocalUser -Name "TestUser" -FullName "Test User" -Description "User for tests"
```

It will then prompt for a password.

To add a user to the local Administrators group, you can run the command `Add-LocalGroupMember` like shown below.

```
Add-LocalGroupMember -Group Administrators -Member TestUser
```

When creating a local Windows user account, you can use the following options below.

- `AccountExpires` – set the expiration date of the account, after which the account will be automatically deactivated (by default, `New-LocalUser` creates an account that never expires)
- `AccountNeverExpires`
- `Disabled` – disable an account after creation
- `PasswordNeverExpires` – set a user’s password to never expire
- `UserMayNotChangePassword` – the user cannot change the account password

You can also manage local user accounts with PowerShell. To list all local Windows users on the current computer, you can use the command `Get-LocalUser`. To get specific information from the users, use the command with the `Select-Object` command. Let's say we want to display all the information of all the users. We can use the command below.

```
Get-LocalUser | Select-Object *
```

To reset a user's password, we can use the command `Set-LocalUser` with a variable that holds the password in a SecureString.

We can add accounts to a newly created local group with the command below.

```
Add-LocalGroupMember -Group 'RemoteSupport' -Member('root','Administrators') -Verbose
```

PowerShell makes it much more efficient to manage users and groups.

File System Management

To view the content of a directory on a Windows file server, you want to use the `Get-ChildItem` cmdlet. The `-Force` flag shows all hidden files. Below, we show all root objects in the Shared folder.

```
Get-ChildItem -Force \\fs\Shared
```

The `-Recurse` parameter checks all subfolders and their content and is added after the folder name. To filter the output, you can use the `Filter`, `Exclude`, `Include`, and `Path` flags to the `Get-ChildItem` cmdlet. You can use the `Where-Object` cmdlet for advanced object filtering.

To create new objects, you can use the `New-Item` cmdlet and specify the type of item you want to create. This could be anything from a directory to a registry key.

```
# makes a new folder
New-Item -Path '\\fs\Shared\NewFolder' -ItemType Directory
# makes a new file
New-Item -Path '\\fs\Shared\NewFolder\newfile.txt' -ItemType File
# creates a file and writes data to it
$text = 'Hello World!' | Out-File $text -FilePath C:\data\text.txt
```

To overwrite an existing file, use the `-Force` switch parameter. You can create a CSV file using the `Export-Csv` cmdlet.

Use the `Remove-Item` cmdlet to delete objects. If the object is not empty, a prompt will appear confirming deletion.

```
Remove-Item -Path '\\fs\Shared\it'
```

If you have already confirmed every object inside the folder should be deleted, you can use the `-Recurse` flag to skip the confirmation step.

To copy objects from one path to another, use the `Copy-Item` cmdlet. You can use the `-Force` flag at the end to overwrite an existing file, even if it is in Read-Only mode. Make sure to use UNC paths if you're copying files to or from remote computers.

```
# creates a backup by copying the file users.xlsx from one remote computer to
another
Copy-Item -Path \\fs\Shared\it\users.xlsx -Destination \\fs2\Backups\it\users.xlsx
```

```
# copy files from local directory to remote folder
Copy-Item C:\data\ -Recurse \\fs\c$\temp
```

You can use the `-Filter` flag to copy only certain files from the source content to the destination.

To move items, you can use the `Move-Item` cmdlet. This includes the item's properties, contents, and child items. It can also move a file or subdirectory to another location.

```
# moves specific backup file from one location to another
Move-Item -Path \\fs\Shared\Backups\1.bak -Destination \\fs2\Backups\archive\1.bak
```

To rename an object without changing it, you can use the `Rename-Item` cmdlet. It is not possible to move the item with this cmdlet.

```
Rename-Item -Path "\\fs\Sahred\temp.txt" -NewName "new_temp.txt"
```

You can rename multiple items at once using a script.

The cmdlet `set-acl` is used to change the security descriptor of a specified item, like a file, folder, or registry key. It modifies file or folder permissions. The `SetAccessRule` parameter completely overwrites the permissions for a user or group. If you want to add permissions, you can use the `AddAccessRule` parameter instead.

```
$acl = Get-Acl \\fs1\shared\sales
$AccessRule = New-Object
System.Security.AccessControl.FileSystemAccessRule("ENTERPRISE\T.Simpson","FullControl","Allow")
$acl.SetAccessRule($AccessRule)
$acl | Set-Acl \\fs1\shared\sales
```

Below you can find other permissions that can be assigned to users or security groups.

Access Right	Access Right's Name in PS
Full Control	FullControl
Traverse Folder / Execute File	ExecuteFile
List Folder / Read Data	ReadData
Read Attributes	ReadAttributes
Read Extended Attributes	ReadExtendedAttributes
Create Files / Write Data	CreateFiles

Access Right	Access Right's Name in PS
Create Folders / Append Data	AppendData
Write Attributes	WriteAttributes
Write Extended Attributes	WriteExtendedAttributes
Delete Subfolders and Files	DeleteSubdirectoriesAndFiles
Delete	Delete
Read Permissions	ReadPermissions
Change Permissions	ChangePermissions
Take Ownership	TakeOwnership

There are sets of basic access rights that can be applied as well.

Access Rights Set	Rights Included in the Set	Name of the Set in PS
Read	List Folder / Read Data Read Attributes Read Extended Attributes Read Permissions	Read
Write	Create Files / Write Data Create Folders / Append Data Write Attributes Write Extended Attributes	Write
Read and Execute	Traverse Folder / Execute File List Folder / Read Data Read Attributes Read Extended Attributes Read Permissions	ReadAndExecute
Modify	Traverse Folder / Execute File List Folder / Read Data Read Attributes Read Extended Attributes Create Files / Write Data Create Folders / Append Data Write Attributes Write Extended Attributes Delete Read Permissions	Modify

To copy permissions, you must own both source and target folders.

```
# copy permissions from "accounting" to "sales"
get-acl \\fs1\shared\accounting | Set-Acl \\fs1\shared\sales
```

You can use the `RemoveAccessRule` parameter to remove permissions. Below we remove the "Allow FullControl" permission for the T.Simpson user to the "Sales" folder. `RemoveAccessRule` only deletes specific permissions. To completely wipe a user's permissions, use the `PurgeAccessRules` command. `PurgeAccessRules` only works with SIDs and with explicit permissions, not inherited ones.

```
$acl = Get-Acl \\fs1\shared\sales
$AccessRule = New-Object System.Security.AccessControl.FileSystemAccessRule("ENTERPRISE\T.Simpson","FullControl","Allow")
$acl.RemoveAccessRule($AccessRule)
$acl | Set-Acl \\fs1\shared\sales
```

We can use the `SetAccessRuleProtection` method to manage inheritance. The first parameter is responsible for blocking inheritance from the parent folder, and it is either true or false. The second parameter determines whether the current inherited permissions are retained or removed, and it is either true or false.

```
# disabling inheritance for "sales" and deleting any inherited permissions
$acl = Get-Acl \\fs1\shared\sales
$acl.SetAccessRuleProtection($true,$false)
$acl | Set-Acl \\fs1\shared\sales
```

You can use the `SetOwner` method to set an owner for a folder. It does not enable you to change the owner to any account you want. The account must have the "Take Ownership", "Read", and "Change Permissions" rights.

```
$acl = Get-Acl \\fs1\shared\sales
$object = New-Object System.Security.Principal.Ntaccount("ENTERPRISE\J.Carter")
$acl.SetOwner($object)
$acl | Set-Acl \\fs1\shared\sales
```

Managing Applications and Software

You can install software with PowerShell in two ways: the standard installation and the silent installation.

Standard installation is the simple installation of the software. You use the `Start-Process` command to perform one or more installations. After that, you need to specify the software path that needs to be executed. It can be seen below.

```
Start-Process C:\Doc\7zip.exe
```

Software installation can also be performed silently. First, we add the `Start-Process` command and specify the software file path. Then we add the `-ArgumentList` parameter and assign `/S /v/qn` options to install the specified software in silent mode. It can be seen below.

```
Start-Process C:\Doc\winRAR.exe -ArgumentList "/s /v/qn"
```

There are two ways to uninstall software using PowerShell, the `Uninstall()` method and the `Uninstall-Package` command. The first method is the easiest to remove well-known programs from a device. The second is good for hidden programs.

First, we can get the list of applications installed on a computer by using the command shown below. You can narrow it down by specifying the object name as the program name, too.

```
Get-WmiObject -Class Win32_Product | Select-Object -Property Name
```

The `Uninstall()` method is built in. All you have to do is call it on your program to uninstall it.

```
# store the program in variable $MyProgram
$MyProgram.uninstall()
```

If PowerShell doesn't list your program, you need to use the `Uninstall-package` command instead.

```
Get-Package -Provider Programs -IncludeWindowsInstaller -Name "NAME"
```

Then you just run this command.

```
Uninstall-Package -Name NAME
```

Active Directory

A **Windows domain** is a group of users and computers under the administration of a given business. The idea behind a domain is to centralize the administration of common components of a Windows computer network in a single repository, called Active Directory.

Active Directory (AD) is Microsoft's directory service or container which stores data objects on your local network environment. The service records data on users, devices, applications, groups, and devices in a hierarchical structure. The structure makes it possible to find the details of resources connected to the network from one location. The server that runs the Active Directory services is known as a **Domain Controller (DC)**.

AD enables users to log on and manage a variety of resources from one location. Login credentials are unified so that it's easier to manage multiple devices without having to enter account details to access every individual machine. All users across the network can be configured with minimum effort and you can configure security policies directly from AD, applying them to users and computers across the network as needed.

How to Setup AD (with RSAT)

To setup AD, you need to have Windows Professional or Windows Enterprise installed so that you can set it up using Remote Server Administration Tools (RSAT).

Method 1:

First, right-click on the `Start` button and go to `Settings > Apps > Manage optional features > Add features`. Select `RSAT: Active Directory Domain Services and Lightweight Directory Tools`. Lastly, select `Install` before going to `Start > Windows Administrative Tools` to access AD once installation is complete.

Method 2:

Make sure the correct version of Server Administrator Tools are installed for your device. Then, right-click the `Start` button and select `Control Panel > Programs > Programs and Features > Turn Windows features on or off`. Click on `Remote Server Administration Tools` and then `Role Administration Tools`. Next, click on `AD DS and AD LDS Tools` and verify `AD DS Tools` has been checked. Press `Ok` and then go to `Start > Administrative Tools` on the `Start` menu to access AD.

How to Set Up A Domain Controller

The first thing you need to do to use AD is to set up a domain controller. A **domain controller** is a central computer that responds to authentication requests and authenticates other computers throughout the network. All other computers connect to the domain controller so that the user can authenticate every device from one location. It stores the login credentials of all other computers and printers.

1. Assign a static IP address to your Domain Controller and install Active Directory Domain Services or AD DS

2. Open `Server Manager` and click `Roles Summary > Add roles and features`
3. Click `Next`
4. Select `Remote Desktop Services Installation` if you're deploying a domain controller in a VM or select `role-based` or `feature-based installation`
5. Select a server from the server pool
6. Select `Active Directory Domain Services` from the list and select `Next`
7. Leave the Features checked by default and press `Next`
8. Click `Restart the destination server automatically if required` and click `Install`. Close the window once complete
9. Once the AD DS role has been installed a notification will display next to the `Manage` menu. Press `Promote this server into a domain controller`
10. Now click `Add a new forest` and enter a `Root domain name`. Press `Next`.
11. Select the `Domain functional level` you desire and enter a password into the `Type the Directory Services Restore Mode (DSRM password)` section. Click `Next`.
12. When the DNS Options page displays click `Next`.
13. Enter a domain in the `NetBios Domain name` box (usually the same as root domain name). Press `Next`.
14. Select a folder to store your database and log files. Click `Next`.
15. Press `Install` to finish. The system will reboot.

It is always a good idea to have at least two domain controllers in case one goes down. Below are instructions for how to add a domain controller to an existing domain in AD.

1. Open `Server Manager`, click on the `Manage` option on the menu ribbon and select `Add Roles and Features`.
2. In the opening screen of the wizard, click on `Next`.
3. In the `Installation Type` screen select the `Role-based or feature-based installation` radio button and click on `Next`.
4. In `Server Selection` leave the only server in the list highlighted and press `Next`.
5. In the `Server Roles` screen, Check the `Active Directory Domain Services` box. A dialogue box appears. Click on the `Add Features` button.
6. Back in the main feature selection screen, click the `Next` button.
7. This cycles through to the `Features` screen. Just click on the `Next` button. In the `AD DS` screen, click on the `Next` button.
8. Finally, click the `Install` button. Once the installation process finishes, you will see a notice telling you that additional steps are required. Click on the link that says `Promote this server to a domain controller`. This brings up the `Deployment Configuration` screen.

9. Leave the `Add a domain controller to an existing domain` radio button active. At the bottom of the list of options, you will see `no credentials provided`. Click on the `Change` button next to that.
10. Enter the username and password of the Administrator account on the AD instance that you first set up. This username should be in the format `<domain>\Administrator`. Click `OK`.
11. On return from the login popup, you will see that the `Domain` field has been populated with the domain that you entered for the user account. Click on the `Next` button.
12. Decide whether to make this a read-only domain controller (RODC). If so, check that box in the `Options` screen, if not, check both the `DNS server` and `Global Catalogue` boxes.
13. Enter a `DSRM password` and confirm it. Click on the `Next` button. You will see a warning but just click on the `Next` button again.
14. In `Additional Options` choose your original domain controller for the `Replicate from:` field. Click on `Next`.
15. Leave all of the paths in their default settings and click on `Next`. In the `Review Options` screen, click `Next`.
16. The system will perform a prerequisites check. If that succeeds, click the `Install` button.
17. The system will reboot after installation.

AD Users

Users are one of the most common object types in AD, and are one of the objects known as **security principals**, meaning that they can be authenticated by the domain and assigned privileges over **resources** like files. Basically, a security principal is an object that acts upon resources in a network.

Users can represent two types of entities: people and services. **People** generally represent any persons in your organization that need access to the network. **Services** require users to run, but the service users are different from regular users as they only have privileges needed to run their specific service.

Machines are another object type in AD. For every computer that joins the AD domain, a machine object will be created. They are also considered security principals and are assigned an account like any regular user, usually having somewhat limited rights within the domain itself. Machine accounts themselves are local administrators on the assigned computer.

Just like regular Windows, you can define and assign user groups certain access rights to resources. Security groups are also considered security principals and have certain privileges over resources on the network. Groups can have both users and machines as members, and if needed, they can include other groups as well.

In the table below, we show some of the most important groups in a domain.

Security Group	Description
Domain Admins	Have administrative privileges over the entire domain; by default they can administer any computer on the domain
Server Operators	Can administer Domain Controllers; cannot change any administrative group memberships
Backup Operators	Have access to any file, ignoring their permissions; used to perform backups of data on computers
Account Operators	Can create or modify other accounts in the domain
Domain Users	All existing user accounts in the domain
Domain Computers	All existing computers in the domain
Domain Controllers	Includes all existing DCs on the domain

In the Domain Controller, we can configure users, groups, or machines in AD. In the start menu, you can search `Active Directory Users and Computers`. By clicking on it, you'll see a window that shows the hierarchy of users, computers, and groups that exist in the domain. These objects are organized in **Organizational Units (OUs)** which are container objects that allow you to classify users and machines. They're mostly used to define sets of users with similar policing requirements.

There are a couple default containers created by Windows automatically. **Builtin** contains default groups available to any Windows host. **Computers** contain any machine joining the network. It's automatically put here by default but you can move them if needed. **Domain Controllers** is the default OU that contains the DCs in your network. **Users** contains default users and groups that apply to a domain-wide context. Lastly, **Managed Service Accounts** holds accounts used by services in your Windows domain.

Security Groups vs OUs

OUs are useful for applying policies to users and computers, including specific configurations that pertain to sets of users depending on their role. A user can only be a member of a single OU at a time.

Security Groups are used to grant permissions over resources. A user can be part of many groups, which is needed to grant access to multiple resources.

To delete an OU, we need to enable the `Advanced Features` in the View menu. This will show additional containers and enable you to disable the accidental deletion protection. Right-click the OU and go to `Properties`. You will find a checkbox in the `Object` tab to disable the protection.

Delegation is a process where you can give specific users some control over some OUs. It allows you to grant specific privileges to perform advanced tasks on OUs without needing a Domain Administrator to step in. You can right-click the OU and select `Delegate Control`. A new window will open and you can add users to whom you want to delegate control. After adding the names, you can select specific tasks for them.

You can also use PowerShell to manage your AD, and more information is found in source 2 on this.

AD Computers

All machines that join a domain (except for DCs) are put in the container called "Computers." It's better to split up your devices from that folder so that you can create different policies for your servers and machines that regular users use. There is no specific rule on how to organize your machines, but it can be a good starting point to divide them by their use.

Workstations are what each user in the domain likely logs in to. This is the device users do their work or normal browsing activities on. They should never have a privileged user signed into them.

Servers are used to provide services to the users or other servers.

Domain controllers allow you to manage the AD Domain. These devices are often deemed the most sensitive devices within the network as they contain hashed passwords for all user accounts within the environment.

To divide these up, you can create OUs (Organizational Units) for Workstations and Servers (Domain Controllers are already in an OU created by default). Then, you move the personal computers and servers to their respective OU. Now you can configure policies for each OU.

Group Policies

We can manage configurations and security baselines to users through **Group Policy Objects (GPO)**. They are a collection of settings that can be applied to OUs and can contain policies aimed at either users or computers, allowing you to set a baseline on specific machines and identities.

We can use the **Group Policy Management** tool from the start menu to configure GPOs. You first have to create a GPO under **Group Policy Objects** and link it to the GPO where you want the policies to apply. Any GPO will apply to the linked OU and any sub-OUs under it.

When selecting a GPO, you'll see its **scope**, which is where the GPO is linked in the AD. You can also apply **security filtering** to GPOs so that they are only applied to specific users/computers under an OU. By default, they apply to the **Authenticated Users** group, which includes all users/PCs.

The **settings** tab includes the actual contents of the GPO and lets us know what specific configuration it applies. The **Default Domain Policy** indicates really basic configurations that should apply to most domains, including password and account lockout policies.

To edit a policy, right-click the GPO and select `Edit`. A new window will open and we can navigate and edit all the available configurations.

GPOs are distributed to the network via a network shared called `SYSVOL` stored in the DC. It may take some time for all the computers to update with a new change to the GPOs, so you can force it to sync immediately by running the command `gpupdate /force`.

Monitoring AD Events

The table below shows some of the most important network events that you should be on the look out for.

Current Windows Event ID	Legacy Windows Event ID	Description
4618	N/A	A security event pattern has been recognized
4649	N/A	A replay attack was detected (potentially a false positive)
4719	612	A system audit policy was changed
4765	N/A	SID History added to an account
4766	N/A	The attempt failed to add SID History to an account
4794	N/A	Attempt to launch Directory Services Restore Mode
4897	801	Role separation enabled
4964	N/A	Special groups have been assigned a new logon

Current Windows Event ID	Legacy Windows Event ID	Description
5124	N/A	Security updated on OCSP Responder Service
N/A	550	Potential DoS attack
1102	517	Audit log was cleared

AD Forests and Trees

A **tree** is an entity with a single domain or group of objects that is followed by child domains. A **forest** is a group of domains put together. When multiple trees are grouped together, they become a forest.

Trees in the forest connect to each other through a **trust relationship**, which enables different domains to share information. All domains trust each other automatically, allowing you to access them with the same account information you used on the root domain.

Each forest uses one unified database, and it sits on the highest level of the hierarchy with the tree sitting at the bottom. One challenge network administrators face is managing these forests and keeping the directory secure.

There are different types of forest designs for a network administrator to choose. Two examples are a single forest design and a multi-forest design. A single-forest design is easier to manage due to its simplicity of comprising the entire network. A multi-forest design divides the network into different forests, increasing security but complicating administrative tasks.

Trust Relationships and Types

Trusts are used to facilitate communication between domains, enabling authenticating and access to resources between two entities. These can be one-way or two-way. Within the trust, the two domains are divided into a trusting domain and a trusted domain.

In a one-way trust, the trusting domain accesses the authentication details of the trusted domain so that the user can access the resources from the other domain. For a two-way trust, both domains accept the other's authentication details. All domains within a forest automatically trust each other, but you can set up trusts between domains in different forests to transfer information.

Trusts can be created through the configuration wizard `New Trusts Wizard`. You can see the `Domain Name`, `Trust Type`, and `Transitive` status of existing trusts and select the type of trust you want to create.

Trust Type	Transit Type	Direction	Default?	Description
Parent and child	Transitive	Two-way	Yes	Established when a child domain is added to a domain tree
Tree-root	Transitive	Two-way	Yes	Established the moment a domain tree is created within a forest
External	Non-transitive	One-way or two-way	No	Provides access to resources in a Windows NT 4.0 domain or a domain located in a different forest that isn't supported by a forest trust
Realm	Transitive or non-transitive	One-way or two-way	No	Forms a trust relationship between a non-Windows Kerberos realm and a Windows Server 2003 domain
Forest	Transitive	One-way or two-way	No	Shares resources between forests
Shortcut	Transitive	One-way or two-way	No	Reduces user logon times between two domains within a Windows Server 2003 forest

once done with website stuff, go back to the pdf for more commands on the AD side that's at the beginning

Sources

1. What is Active Directory? A step-by-step tutorial
2. Windows PowerShell Tutorial for Beginners
3. Active Directory Basics
4. How to Create, Change, and Remove Local Users or Groups with PowerShell
5. How to Uninstall Software Using PowerShell
6. Install Software Using PowerShell Script