JUNE 2024

CREATING AN ACTIVE DIRECTORY DOMAIN
JOINING AN AD DOMAIN W/ WINDOWS 10
DEPLOYING CIS BENCHMARKS VIA GROUP POLICY

BLAKE NIEBRUGGE
SECURITY ANALYST

Table of Contents

Abstract

This document details the self-learning exercise I completed, which involved configuring and creating a new Active Directory (AD) forest from scratch using a Windows Server 2019, promoting it to a domain controller, joining the domain with a Windows 10 workstation, and deploying CIS Benchmarks to the workstation via Group Policy. This entire process was conducted using trial versions of Windows and free tools, which are detailed later. I undertook this project as I was curious on the construction of enterprise networks and how to manage them. For any questions or comments, please contact me at the email provided below.

Blake Niebrugge
Security Analyst

blakeniebrugge.com
blake@blakeniebrugge.com

Hardware Specs/Software Used

Hardware

Dell OptiPlex 3040 (micro-form)

- Intel Core i5-6500T CPU @ 2.5 GHz (4 cores)
- 8 GB RAM
- 500 GB Storage (Samsung EVO SSD)
- Linux Ubuntu OS (22.04.4 LTS)

Software/Packages

Oracle VM VirtualBox v7.0

Windows Server 2019 ISO

Note: This can be obtained for free from Microsoft's Evaluation Center. It allows for a free 180 trial period without a product key.

Windows 10 Pro 22H2 ISO

Note: This ISO can be downloaded for free here – there aren't very many restrictions running an inactivated version of Windows.

Setting Up Active Directory

To begin, I started with Oracle's VirtualBox. This is completely free and can be found here (at the time of writing, this was done with v7.0). There's no other add-ins or packages needed for that. Once that was installed, I started with building the server virtual machine. First, I had to get a disk image to deploy on VirtualBox. Microsoft has an evaluation center for its line of server operating systems, which allows for 180 days of unrestricted use. Following the link found in "Software Used" above, I downloaded the Windows Server 2019 ISO.

Setting up the VM in VirtualBox

Once we get the ISO, we created a new VM in VirtualBox with the following parameters:

Name: ADDS
*Folder: *Default Path**
*ISO Image: Browse > *Find newly downloaded .iso file"*
Type: Microsoft Windows
Version: Windows Server 2019 Evaluation (Desktop Experience)

User: BDNadmin
*Password: *************
*Product Key: *Leave as is**
*Hostname: *Leave as is**
*Domain: *Leave as is**

Once past that, I assigned the VM 4GB of RAM and 2 vCPUs. It's not a lot, but since this domain is only going to include one other machine and a couple users, we don't need actual enterprise-grade performance to run it. Once that was assigned, I provisioned 64GB of virtual storage to the VM as well. Same reason as the RAM/vCPUs. I also had to keep in mind the physical restrictions we have with our Linux host, we can't assign more RAM/vCPUs/Storage than we actually have. I ran into this issue on the workstation side with over-assigning RAM, which I will discuss in the "Problems Encountered" section towards the end.

After the server VM was created, I went back into the VM's settings on VirtualBox to change a couple things. First, after the VM was mounted on the virtual hard drive and running, I disabled the floppy disk option. VirtualBox had issues trying to boot from the virtual floppy for the Windows 10 image, so I disabled it completely for the server too while I was there. After that, I went to the networking settings and instead of using NAT as the networking configuration, I changed it to "Bridged Adapter". With NAT, VirtualBox blocks incoming connections by default, making it a little difficult to run an AD server. A Bridged Adapter circumvents VirtualBox and the Host machine's network stack, essentially putting the VM on the same network as everything else on my home Wi-Fi. Once that was done, I was finally able to get into the server and start setting the OS up.

Setting up Windows Server 2019 & AD DS

Once the server was up and running, the first thing I needed to do was give it a static IP address. I went to *Settings* > *Network & Internet* > *Change Adapter Settings* > *Ethernet* > *Internet Protocol Version 4 (TCP/IPv4)*. I gave it the following configuration:

> *Use the Following IP Address:*
> *IP address: 10.0.0.11*
> *Subnet Mask: 255.255.255.0*
> *Default Gateway: 10.0.0.1*
>
> *Use the following DNS server addresses:*
> *Preferred DNS Server: 9.9.9.9*
> *Alternate DNS Server: *Empty**

There is no specific reason for choosing 10.0.0.11, it's just an arbitrary address I knew was open on my home network. 10.0.0.1 is the IP of my home router and 9.9.9.9 is just a well-known public DNS server. Once this was set, I pinged the newly provisioned static IP from my phone (using iSH, a very cool app if you want a command line interface on your iPhone) and I was getting packets back successfully. That's basically all we needed before setting up AD DS.

In Server Manager, I went to *Manage* > *Add roles & features,* and used these configurations for each of the prompt sections:

Before you begin: Leave as is.
Installation Type: Selected *Role-based or Feature-based Installation*.

Server Selection: Make sure "ADDS" (our server name) is selected.
Server Roles: Select *Active Directory Domain Services*, then *Add Features* on the resulting pop-up.
Features: Leave as is.
AD DS: Leave as is.
Confirmation: Checkmark *Restart the destination server automatically if required*, then *Yes* on the resulting pop-up, select Install. This installs AD DS, Group Policy Management, and some other administration tools. This also configures the DNS server on the machine, which we will need as well.

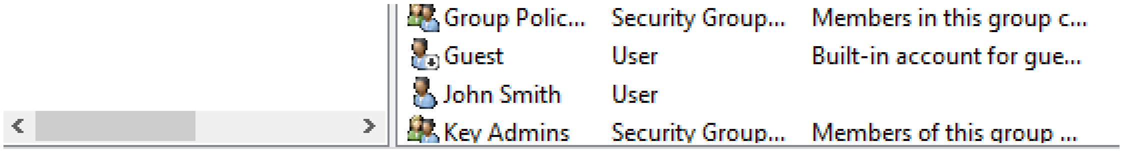
Once all that is downloaded, there is an option on the confirmation screen to promote the server to a domain controller. I selected that and went through the process of setting up our new AD domain, looking similar to how we installed AD DS above:

Deployment Configuration: Select *Add a new forest*, and I named my domain "blakeniebrugge.com" (creative, right?)
Domain Controller Options: Left everything as is, but set a password for the Directory Services Recovery Mode.
DNS Options: Leave as is.
Additional Options: Verify NetBIOS name.
Paths: Leave as is.
Review Options: Verify options.
Prerequisites Check: Once verified, select Install.

The server will restart automatically once installed. After this, we have officially created our AD forest/network and are ready to add users, join computers, etc.

Add a new User to AD

Now that our AD is off the ground, we will need to add a user that will be able to log into the workstation we eventually add to the domain. This part is probably the easiest of all. In Server Manager, we go to *Tools > Active Directory Users and Computers.* We then go to the default "Users" folder, right click > New > User. Our first user was John Smith. I don't know any John Smiths. All I did in this dialog was enter his first and last name, gave him the username john.smith@blakeniebrugge.com (arguably one of the best naming conventions you can have for an enterprise network in my opinion), and gave him a password that won't expire. I also put him in the built-in Administrators group since he would need those privileges in the workstation we add on the next step.



Group Polic...	Security Group...	Members in this group c...
Guest	User	Built-in account for gue...
John Smith	User	
Key Admins	Security Group...	Members of this group ...

At this point I now had a domain controller, an AD forest/network, and our first user. To test the capabilities of this new network, I now needed a workstation that we could log into using the John Smith credentials.

Joining Active Directory with a Windows 10 Workstation

Setting up the VM in VirtualBox

We are now at the point where I need to set up the workstation VM. Initially I was going to use a Windows 11 image to do this, but the minimum hardware specs called for 4 GB of RAM – 4 GB I don't physically have after the DC's allocation plus my Host's usage. I did end up trying to make it work, but it kept crashing the DC and freezing the Host. More to come in the "Problems Encountered" section. On the other hand, Windows 10 had pretty lenient hardware requirements - only needing 1 GB of RAM, so that's what I ended up going with. In VirtualBox we started creating the workstation VM much like we did the server VM.

Name: Workstation1
*Folder: *Default Path**
*ISO Image: Browse > *Windows 10 .iso file**
Type: Microsoft Windows
Version: Windows 10 Pro

User: BDNadmin
*Password: ************
*Product Key: *Leave as is**
*Hostname: *Leave as is**
*Domain: *Leave as is**

Once created, I followed the same steps in the VM settings on VirtualBox as I did the server – disabling the floppy drive and configuring the network option to bridge to the Host adapter. I probably could have left NAT networking on but I didn't want to have to deal with the issues if it didn't work. I also had to delete any answer files the ISO created/copied over since Windows would try to try to find a product key there and wouldn't boot if it couldn't find it.

Setting up Windows 10 Pro and Joining the Domain

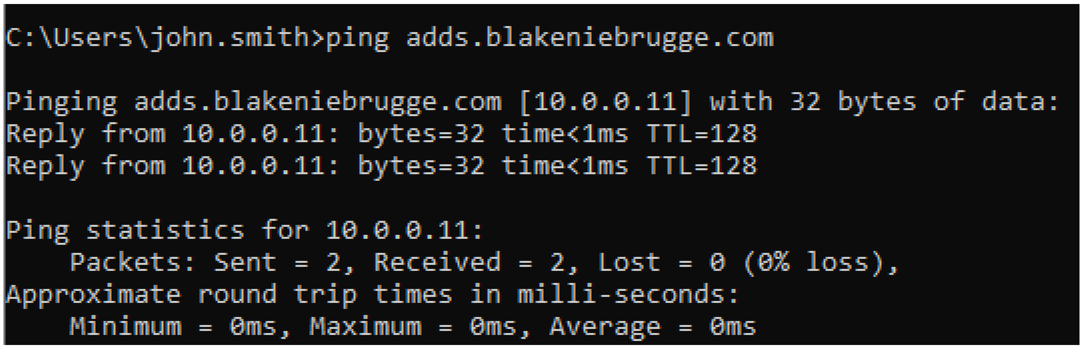
Once we start up the VM we go through a much lengthier set up with the OS than we did on the server. First, we skip the product key page since we don't have one. After that, we select Custom install since we aren't upgrading an existing OS. We select the only drive listed to install the

OS onto, then I am ready to actually configure the OS. We select the basics like language and keyboard layout, then “Set up for an Organization”. Instead of logging in with a Microsoft account, we choose “Domain join instead”. This lets us simply set up a local admin account from the beginning. We give it a basic name and password since it’s going to change anyway, and then turn off or skip all personalization options presented to us. We are now ready to get to the more technical setup.

At this point I created configured a static IP and pointed DNS to our new DC so it can pick up our domain. I went to *Settings > Network & Internet > Change Adapter Settings > Ethernet > Internet Protocol Version 4 (TCP/IPv4)* and set the following:

> *Use the Following IP Address:*
> *IP address: 10.0.0.12*
> *Subnet Mask: 255.255.255.0*
> *Default Gateway: 10.0.0.1*
>
> *Use the following DNS server addresses:*
> *Preferred DNS Server: 10.0.0.11*
> *Alternate DNS Server: *Empty**

To make sure we could pick up the DC, I pinged it from command line and was able to see packets getting returned. *Note: This screen clip was taken after the domain join*

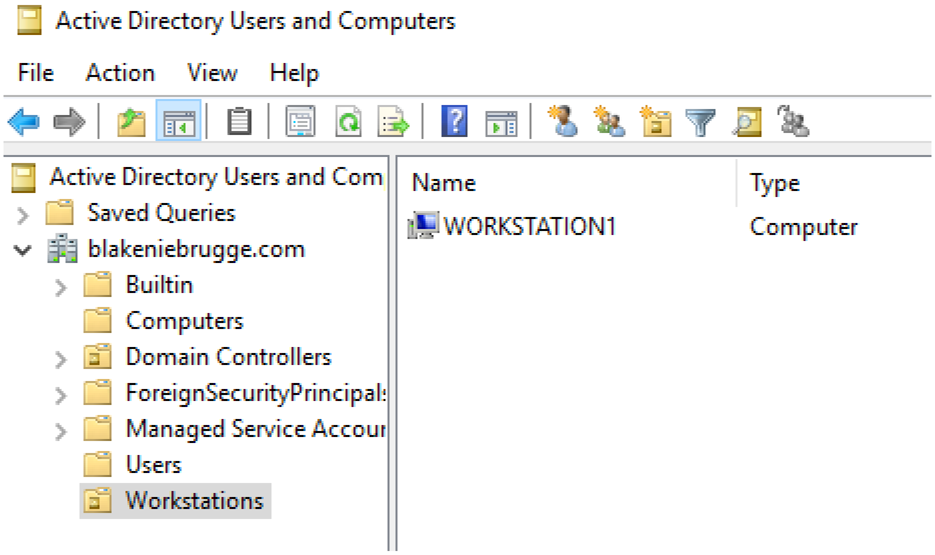


```
C:\Users\john.smith>ping adds.blakeniebrugge.com

Pinging adds.blakeniebrugge.com [10.0.0.11] with 32 bytes of data:
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.11:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

After this, I went to *Control Panel > searched “Join a Domain” > Join a Domain.* On this dialog, I renamed the computer “Workstation1” to match the VirtualBox name, and joined blakeniebrugge.com. After entering John Smith’s new credentials, I was greeted by a dialog that welcomed me to the blakeniebrugge.com domain! The VM restarted and was ready for credentials from our domain. Just to be verify everything was working properly, I created a new user – which was myself (blake.niebrugge@blakeniebrugge.com) – and tried logging into the workstation. Everything was working as expected.



Name	Type
WORKSTATION1	Computer

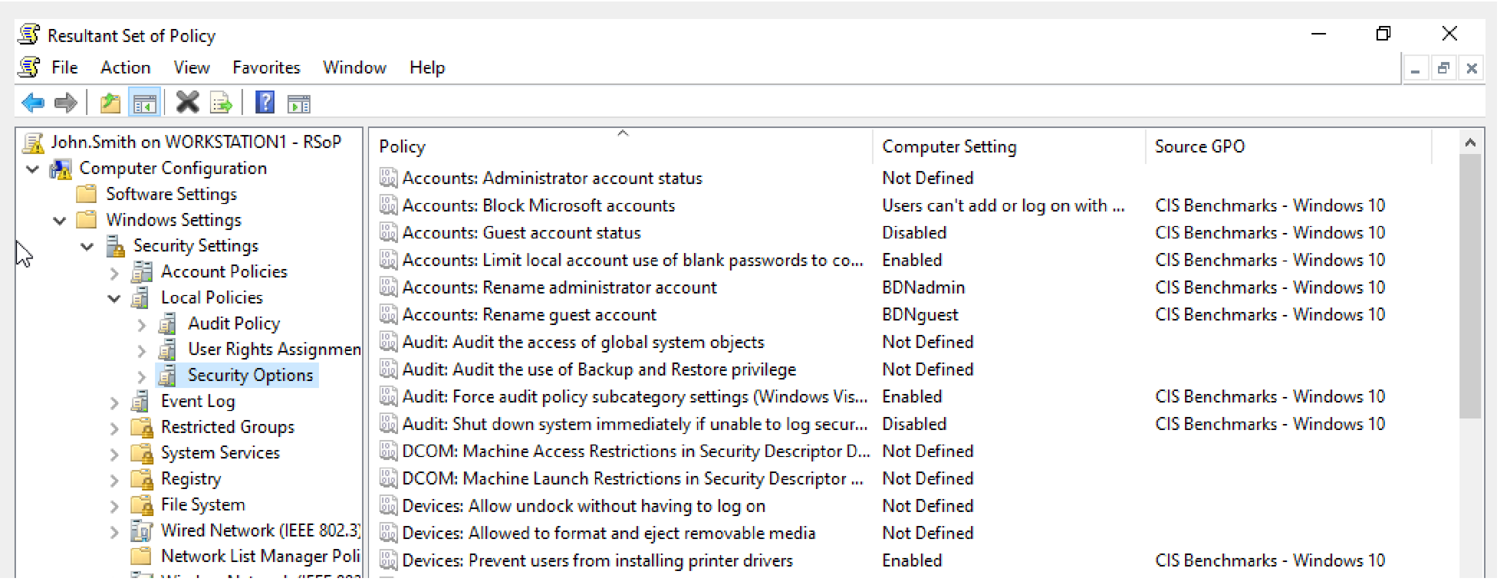
Now I wanted to reconfigure the workstation as if it was getting deployed to an end user. My first idea was following the CIS benchmarks for Windows 10, and that leads into our next section.

Deploying CIS Benchmarks via Group Policy

With the AD set up, including a couple new users and a Windows 10 machine, I proceeded to explore Group Policy Management listed under the Tools on Server Manager. To ensure that the workstation met some sort of enterprise environment standard, I decided that I would deploy the well-known CIS benchmarks via GPO.

I utilized the Windows 10 Enterprise v3.0.0 benchmarks, which encompass all Level 1 (L1), Level 2 (L2), BitLocker (BL), and some Next Generation (NG) configurations. This comprehensive set included around 500 configurations to be deployed. With the official documentation from CIS, navigating through the GPM console to locate each setting was fairly straightforward, albeit time-consuming, taking a few hours to complete.

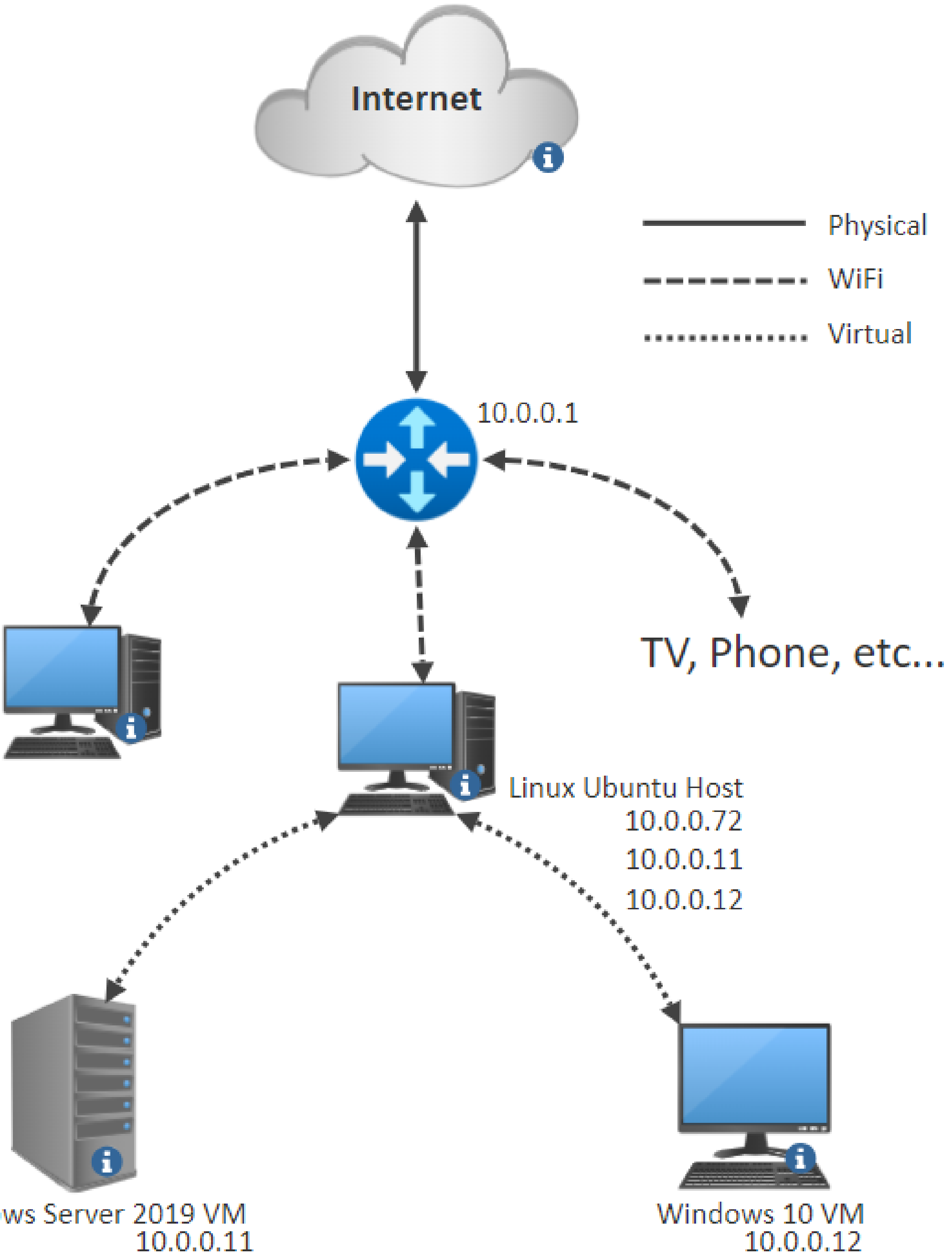
Initially, I mistakenly applied the policies domain-wide, an issue I will talk about in the "Problems Encountered" section. After correcting this, I verified that the workstation successfully received the policies, completing the deployment process. We now had a workstation was domain joined, and was set up to be as secure as CIS deems possible. *Pictured below is the resultant set of policy found on the workstation, verifying that it was indeed receiving the GPO.*



Policy	Computer Setting	Source GPO
Accounts: Administrator account status	Not Defined	
Accounts: Block Microsoft accounts	Users can't add or log on with ...	CIS Benchmarks - Windows 10
Accounts: Guest account status	Disabled	CIS Benchmarks - Windows 10
Accounts: Limit local account use of blank passwords to co...	Enabled	CIS Benchmarks - Windows 10
Accounts: Rename administrator account	BDNadmin	CIS Benchmarks - Windows 10
Accounts: Rename guest account	BDNguest	CIS Benchmarks - Windows 10
Audit: Audit the access of global system objects	Not Defined	
Audit: Audit the use of Backup and Restore privilege	Not Defined	
Audit: Force audit policy subcategory settings (Windows Vis...	Enabled	CIS Benchmarks - Windows 10
Audit: Shut down system immediately if unable to log secur...	Disabled	CIS Benchmarks - Windows 10
DCOM: Machine Access Restrictions in Security Descriptor D...	Not Defined	
DCOM: Machine Launch Restrictions in Security Descriptor ...	Not Defined	
Devices: Allow undock without having to log on	Not Defined	
Devices: Allowed to format and eject removable media	Not Defined	
Devices: Prevent users from installing printer drivers	Enabled	CIS Benchmarks - Windows 10

Topology

This is how the network was laid out from a visual perspective. The VMs were set up with bridged network adaptors, essentially putting them on my home network off the same adapter as the host PC.



Problems Encountered

Note: These are in no particular order

- **Problem:** Domain Controller was unreachable from Workstation1, preventing domain joining.
    - **Solution:** In the VirtualBox settings for the DC, I flipped the Networking method from NAT to Bridged Adapter. I was then able to ping the DC from my own phone and Workstation1. I was then able to join the domain once the workstation could resolve “blakeniebrugge.com”.
- **Problem:** Couldn’t run Windows 11 without crashing Host, or the DC VM.
    - **Solution:** Not really a solution, but I found that Windows 11 was eating up all the RAM I had available on my Host. Giving the DC 4GB and also the Windows 11 VM 4GB left nothing for my Host to run. Hardware requirements for Windows 11 call for 4GB, so I decided to go back to Windows 10 as it only called for 1GB RAM in its requirements. I was able to set up Windows 10 with only 2GB RAM, and everything was working fine again.
- **Problem:** Group Policy Management was no longer available during the deployment of the CIS benchmark GPO.
    - **Solution:** As it turns out, during my first attempt at deploying CIS baselines, I made the GPO domain-wide rather than specific for the workstation. I suspect when I turned off the “Server” system service as a part of the recommendations, the DNS server turned off, resulting in GPM no longer being able to locate/resolve the DC (like being unable to find the front door of your own house) and essentially locking me out of the console. I tried going to SYSVOL to edit and also to remove the policy, to no avail. I rebuilt the DC, and started over with the GPO. I added the workstation to a new “Workstations” Organizational Unit, and linked the GPO to that (You can see the OU on the screenshot after joining the domain above). I slowly started going through the recommendations again, verified the DC was not receiving the updates while the Workstation was, and we were back on track. I probably could have recovered from this, but with such a small AD, it would have taken longer to figure that out than simply rebuilding it.

Conclusion and What's Next

Completing this self-learning exercise has provided valuable insights into the process of setting up and managing an AD environment from scratch. By building a Windows Server 2019 as a domain controller, adding users, and joining a Windows 10 workstation to the domain, I have gained valuable hands-on experience in the fundamental tasks associated with enterprise network administration. Deploying CIS Benchmarks via Group Policy further enriched my understanding of security best practices and configuration management within an AD environment. The challenges encountered, such as managing hardware resource allocation and resolving network configuration issues, provided practical problem-solving experience as well.

As for what's next, this set up really opened the door for a few different possibilities. My initial ideas were along the lines of:

- AD replication and failovers/Server recovery
- BitLocker use
- Active Directory Federation Service/SSO
- Penetration Testing
- More to come…

As stated before, feel free to reach out with any questions/comments/concerns to my email on the "Abstract" page, and I will be sure to get in touch.