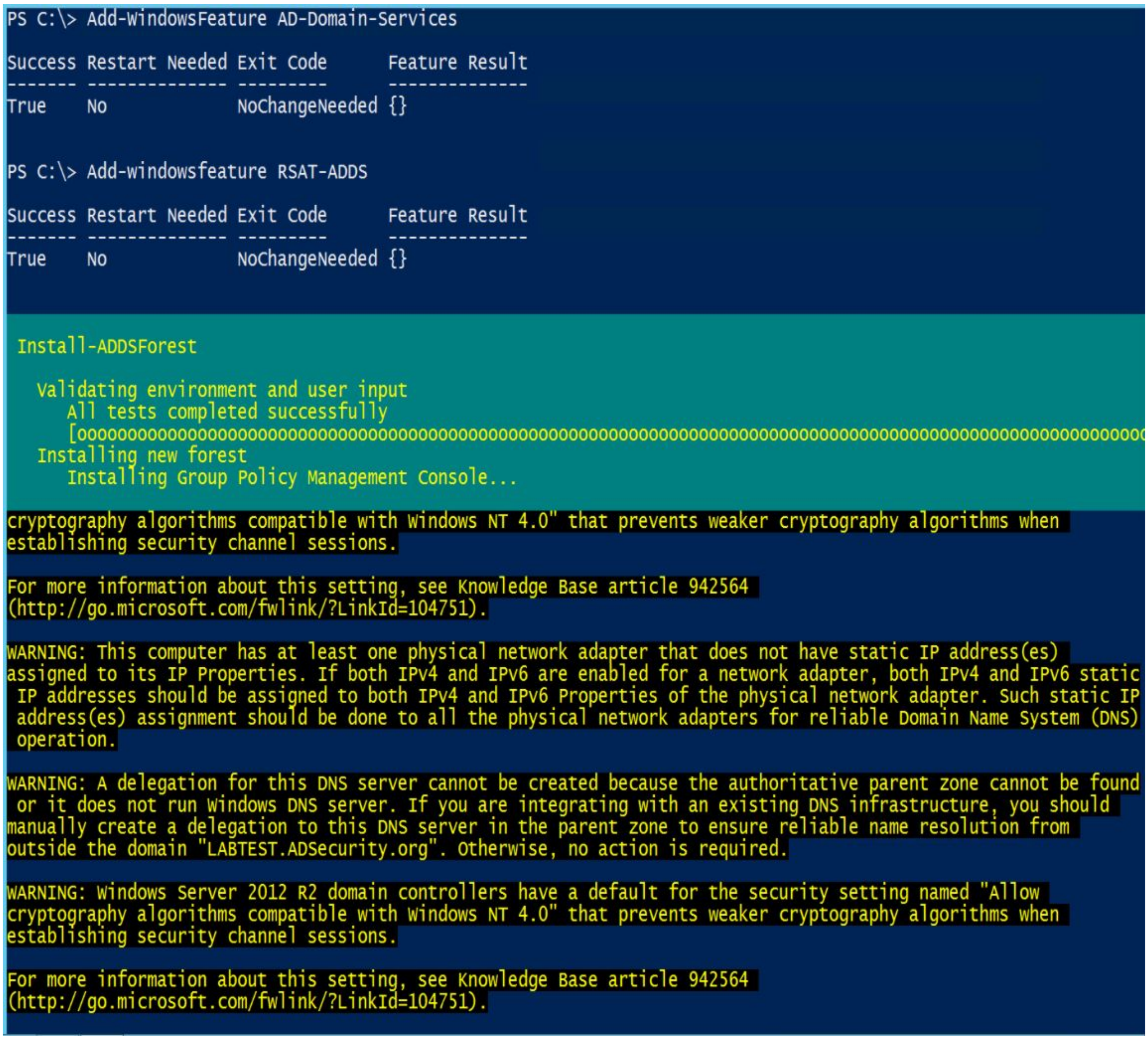
Building an Effective Active Directory Lab Environment for Testing

adsecurity.org

Sean Metcalf

February 23, 2016



```
PS C:\> Add-WindowsFeature AD-Domain-Services

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             NoChangeNeeded {}


PS C:\> Add-windowsfeature RSAT-ADDS

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             NoChangeNeeded {}


 Install-ADDSForest

   Validating environment and user input
      All tests completed successfully
      [oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
   Installing new forest
      Installing Group Policy Management Console...
cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when
establishing security channel sessions.

For more information about this setting, see Knowledge Base article 942564
(http://go.microsoft.com/fwlink/?LinkId=104751).

WARNING: This computer has at least one physical network adapter that does not have static IP address(es)
assigned to its IP Properties. If both IPv4 and IPv6 are enabled for a network adapter, both IPv4 and IPv6 static
 IP addresses should be assigned to both IPv4 and IPv6 Properties of the physical network adapter. Such static IP
 address(es) assignment should be done to all the physical network adapters for reliable Domain Name System (DNS)
 operation.

WARNING: A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found
 or it does not run Windows DNS server. If you are integrating with an existing DNS infrastructure, you should
manually create a delegation to this DNS server in the parent zone to ensure reliable name resolution from
outside the domain "LABTEST.ADSecurity.org". Otherwise, no action is required.

WARNING: Windows Server 2012 R2 domain controllers have a default for the security setting named "Allow
cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when
establishing security channel sessions.

For more information about this setting, see Knowledge Base article 942564
(http://go.microsoft.com/fwlink/?LinkId=104751).
```

This post is not meant to describe the ultimate lab configuration. Instead the focus is on a lab environment that can be stood up quickly and easily as a learning tool. The best way to learn about computer networking and security is to have a home lab. The great thing is that a home lab no longer requires several physical computers as it did in the past. Virtualization enables anyone to take a computer with a decent processor and enough RAM to create a lab environment without being overly complex. Furthermore, it’s possible to build a Windows environment at minimal cost for testing.

Hosting The Lab

The Cloud:

Amazon AWS, Microsoft Azure, and others provide capability to install and configure VMs in the cloud which is helpful when traveling since the lab is available and accessible from anywhere (perhaps saving power at home).

The Server:

I have friends that buy older servers from various internet sources (ebay, etc) at a tremendous discount and run those with (potentially) massive hard drive arrays. The big drawback is the power consumption (and associated power bill). The associated components are usually more expensive, though they do last longer.

The Workstation:

This is my preference – build/buy a hefty workstation-class system with a Core i7 processor. I highly recommend using an SSD as the primary OS drive. Also highly recommended is using a separate SSD for the Virtual Machine files. SSDs are exponentially faster than traditional hard drives and the difference is obvious when running a lab on them. For example, my lab computer has 2 SSDs: a C: drive and a D: drive. I can build a new VM in ~7 minutes. Installing a new Windows Server from an ISO file on the C: drive (SSD) takes ~12 minutes. Also, the server VMs boot almost instantly! It’s extremely fast!

The key is to outfit the lab computer with as much RAM as possible. My recommendation is 16GB at a minimum, 32GB preferred, with more than that even better!
What matters in the system:

- **Processor**: Does the work for the virtualization host as well as all VMs. Core i7 (or better) preferred.
- **Hard Drive**: SSD all the way! Recommend at least 128GB for system drive and at least 256GB for the drive holding the VM files (preferably more!). I also use a traditional hard drive 1-3TBs in size for VM backups. I really like the Samsung EVO SSDs since they are fast and reliable. A 500GB Samsung EVO SSD runs around $300 online (possibly cheaper by the time you read this).
- **Memory**: This is the one you want to put your money into. Personally, I would rather spend a little bit more upfront and have the ability to put 64GB (or more) into a system, then go cheap and have the computer max out at 16GB. The more memory you have, the more VMs you can run which means you can run more involved (& interesting!) scenarios.

I also attach external traditional hard drives (1.5TB and larger) for lab VM backups, though I tend to keep the operating system ISO files and OS template VM files (Sysprep’d operating system VMs) on a SSD for maximum install speed.

Virtualization

There are several virtualization platforms available on the market. Given that my career has involved the Microsoft platform and ecosystem, Hyper-V is the obvious choice for me. VMWare is the 100lb gorilla in this space, so if you are not Windows focused, this may be the best route for you. Note that both Hyper-V and VMWare have a free* virtualization offering: Hyper-V Server and VMWare vSphere Hypervisor (formerly VMWare ESXi). There are others like VirtualBox that install on top of an existing OS.

* "Free" is relative and you should double-check what the license actually means. I Am Not A Lawyer (IANAL) and this information is free, so no guarantees are given.

Active Directory Lab Environment Options

Obviously there are many more options than the few I describe here, but I want to call these out to help those trying to figure out what's best for them.

I use Windows 7 on the client (workstation) if I am not testing something Windows 8 or Windows 10 specific. I tend to run two Domain Controllers (DCs), one running Windows 2008 R2 and the other Windows 2012 R2 to test specific OS issues. Beta software is segmented in its own environment to isolate beta issues.

Basic Active Directory Lab Environment Setup:

This is your basic configuration which supports most test scenarios. Most of the security scenarios I test require only a single DC and on client (workstation).

1. VM 1: Windows 2012 R2 – Domain Controller (single DC for 1 domain forest)
2. VM 2: Windows 7/8/10 – Windows workstation joined to Active Directory (I usually go with Windows 7 since it's common in enterprises)

Standard Active Directory Lab Environment Setup:

This is my standard lab configuration which supports an expanded test scenarios. By having one DC running Windows Server 2008 R2 and another DC running Windows Server 2012 R2, it enables testing of two different DC operating systems. I just shut down the one I'm not testing (briefly) and then stand it up again when done.

1. VM 1: Windows 2008 R2 – Domain Controller (DC #1 for 1 domain forest)
2. VM 2: Windows 2012 R2 – Domain Controller (DC #2 for 1 domain forest)
3. VM 3: Windows 7/8/10 – Windows workstation joined to Active Directory (I usually go with Windows 7 since it's common in enterprises)

Expanded Active Directory Lab Environment Setup:

This is my expanded lab configuration which supports an several, more advanced test scenarios. I run three DCs supporting a two domain AD forest as well as two different server operating systems and two different client operating systems.

1. VM 1: Windows 2008 R2 – Domain Controller (DC #1 for root/parent domain forest)
2. VM 2: Windows 2012 R2 – Domain Controller (DC #2 for root/parent domain forest)
3. VM 3: Windows 2012 R2 – Domain Controller (DC #1 for child domain forest)
4. VM 4: Windows 7 – Windows workstation joined to Active Directory root/parent domain
5. VM 5: Windows 7/10 – Windows workstation joined to Active Directory root/parent domain (or child domain depending on testing scenario)
6. VM 6: Windows Server 2008 R2 – “Application” Server joined to root/parent domain.
7. VM 7: Windows Server 2012 R2 – “Application” Server joined to root/parent domain.

Realistically, you can get fancier than this by adding a Read-Only Domain Controller (RODC), configuring DCs in different subnets associated with different sites, adding more OS variety, installing common enterprise applications such as SCCM, SCOM, Exchange, SQL, SharePoint, etc.

Lab VM Operating Systems

Microsoft used to have “TechNet Plus” which provided all operating systems and most Microsoft applications for $350 the first year and $250 for renewals. This was a great deal which Microsoft removed as an option a few years back. Microsoft discontinued TechNet Plus in favor of a cloud approach which is Microsoft’s TechNet Evaluation Center. The TechNet Evaluation Center provides a one-stop shop for testing out Microsoft products (with pre-built virtual labs!) and has product trial versions (typically 180 days) to download. This is your best bet if you don’t have cash to spend on the MSDN options. For more in-depth information on Microsoft products, the Microsoft Virtual Academy is a wealth of resources – think Khan Academy for Microsoft products.

The MSDN comparison chart explains the benefits of each subscription level.

Visual Studio Premium with MSDN is the option that covers everything, providing dev/test versions of software, Visual Studio, Visual Studio Online, Windows Phone & Store dev account, Microsoft Office Professional Plus 2013, and even an Office 365 Developer Subscription. You also get a Windows Azure $100 monthly credit. The issue is that this subscription costs $6,000 for year one and $2,500/year renewal. You get pretty much everything with this option and you should for over 6 thousand dollars the first year. I have heard there are cheaper options which run ~$2500 for 2 years (several friends purchased MSDN from them and had no issues), but as always, buyer beware.

MSDN platforms is a scaled-down, cheaper option (~$1,200/year) that includes OS and most server products (does not include Microsoft Office products). It does also include Visual Studio Online and Windows Azure $100 monthly credit, so this is the most cost-effective option for Microsoft OS & Server dev/test work. If you happen to be a charity, MSDN Platforms drops in price to $342/year [CDW link].

MSDN Operating Systems runs $800/year, though as the name states, it’s ONLY Operating Systems…

If you have a company, are a Microsoft Partner, and pay ~$500 for the Microsoft Action Pack Subscription (MAPS), you get 1 – 10 licenses for client apps (Office, Visio, etc), Windows workstation & server OSs, and Servers (Exchange, SQL, SharePoint). Though the licenses for most Server products are limited to 1 install.

There's a better Microsoft licensing story if you're a student. Check out DreamSpark where you can get a bunch of Microsoft software for free.

All plans and prices are subject to change.

NOTE: If you go the product trial route, the trial version is only good for 180 days. However, running "*slmgr.vbs -rearm*" from a command window as an administrator extends the trial by 30 days. You can "rearm" Windows twice in this manner which means the trial is good for a total of 240 days after which you will have to start over.

Managing the Lab VMs

Most virtualization platforms enable the creation of a virtual "switch" where network communication can be limited and isolated. This ensures that one lab environment can't communicate with the other.

There are two different approaches to lab OS builds. One is to perform a Windows install (client or server os) and then run Sysprep with the generalize option. This sets Windows to a state similar to just after setup has run ensuring any computer specific information is cleared and generated at next bootup. After running Sysprep, shutdown the computer and save the VM file. It can now be used as the "starting point" for any new OS install (same OS version). Licensing is reset by Sysprep (SkipReam doesn't reset the licensing data).

Once you install the VMs and configure them the way you like, it's important to back up that data. How this is done depends on the virtual platform, but in general I find it best to export the VM data for easy import later. I typically backup VMs after making significant changes to the lab environment (my baseline) as well as before running any attacks with "hacker tools."

Don't turn off some Domain Controllers over an extended period of time. Turn them all off or have them all on. If you need to test a scenario which involves turning off all DCs except one or two, make sure they are all on at the same time to replicate and then turn them all off. Murphy's Law: Troubleshooting Active Directory issues in a lab can sometimes seem more complex than in production.

VM Specs

Most lab VMs only require a single virtual CPU (vCPU) and I typically set each virtual hard drive (VHD/VHDX on Hyper-V) to 250GB which dynamically expands (without any real performance issues).

Lab VM Disk Sizes (after install & config):

- Windows 7: 50 – 60GB
- Windows 10: 15 – 25GB
- Windows Server 2008 R2: 7 – 10GB
- Windows Server 2012 R2: 7 – 10GB
- Windows Server 2016: 12 – 15GB

Obviously the VM disk space doesn't approach the 250GB set for the virtual hard drives, but that's simply the maximum size for the VHD. This provides extra space if needed in the VHD.

RAM configuration takes a little more tweaking. Dynamic Memory (Hyper-V & VMWare) simplifies this since you can set the start-up RAM to be something like 2GB (most of my lab VMs are set to 2GB for start-up RAM). After boot-up, the VM OS settles in to a smaller amount. Though I've found that Windows 8/8.1 & Windows 10 do better with 3GB for start-up RAM. Any OS that doesn't have virtualization extensions (VMWare Tools or Hyper-V Integration Services) typically don't support these features, so the RAM will stay the same while the system is powered on (typically non-Windows VMs). Obviously the more memory provided to the VM, the quicker it operates (generally speaking), so don't starve the VMs of memory unless you have to. Most of the Windows VMs I run settle into using around 700MB to 1,500MB of RAM and typically newer Windows operating systems do better with more RAM.

Promoting the Domain Controllers

With Windows Server 2008 R2 and older Windows server operating systems, DCPromo is used to promote a member server to Domain Controller. Starting with Windows Server 2012, the PowerShell cmdlet *Install-ADDSForest* creates a new forest and *Install-ADDSDomainController* is used to add a new DC to an existing domain. *Install-ADDSDomain* creates a new domain in an existing forest.

Install the proper modules:

Add-WindowsFeature AD-Domain-Services

Add-windowsfeature RSAT-ADDS

Create a new forest by creating the first DC in the forest:

Import-Module ADDSDeployment

$SafeModeAdministratorPasswordText = '&P@ssw0rd2013&'
$SafeModeAdministratorPassword = ConvertTo-SecureString -AsPlainText $SafeModeAdministratorPasswordText -Force

Install-ADDSForest -CreateDNSDelegation:$False -DatabasePath "c:\Windows\NTDS" -DomainMode 'Win2012' -DomainName "LAB.ADSecurity.org" -DomainNetbiosName "ADSECURITYLAB" -ForestMode 'Win2012' -InstallDNS:$true -LogPath

```
“C:\Windows\NTDS” -NoRebootOnCompletion:$false -Sysvolpath “C:\Windows\SYSVOL”
-Force:$true -SafeModeAdministratorPassword $SafeModeAdministratorPassword
```

Add another DC to an existing domain:

```
Import-Module ADDSDeployment
```

```
$SafeModeAdministratorPasswordText = ‘&P@ssw0rd2013&’
$SafeModeAdministratorPassword = ConvertTo-SecureString -AsPlainText
$SafeModeAdministratorPasswordText -Force
```

```
Install-ADDSDomainController -NoGlobalCatalog:$false -CreateDNSDelegation:$false -
Credential (Get-Credential) -CriticalReplication:$false -DatabasePath
“C:\Windows\NTDS” -DomainName “LAB.ADSecurity.org” -InstallDNS:$true -LogPath
“C:\Windows\NTDS\Logs” -SiteName “Default-First-Site-Name” -SYSVOLPath
“C:\Windows\SYSVOL” -Force:$true -SafeModeAdministratorPassword
$SafeModeAdministratorPassword
```

Add a new Domain to an existing Forest by creating the first DC in the domain:

```
Import-Module ADDSDeployment
```

```
Install-ADDSDomain -Credential (Get-Credential) -NewDomainName “CHILD” -
ParentDomainName “LAB.ADSecurity.org” -InstallDNS -CreateDNSDelegation -
DomainMode Win2003 -DatabasePath “C:\Windows\NTDS” -SYSVOLPath
“C:\Windows\SYSVOL” -LogPath “C:\Windows\NTDS\Logs”
```

Note: These AD DCPromo PowerShell commands are meant to guide, not to be used verbatim.
Use Get-Help with the desired cmdlet to identify the appropriate parameters.

My “ADSecurity.org” Lab Environment

My lab environment has several different configurations running simultaneously. I have three Hyper-V hosts, each with Core i7, 32GB, and a 256GB SSD. Two of them run Windows Server 2012 R2 and the third runs Windows Server 2016 TP4.

On the two 2012 R2 Hyper-V hosts, I run several lab AD environments to test out a variety of test scenarios (including variations of what I’ve described above). Typically one Hyper-V host contains my “stable” lab with long term VM testing (including Microsoft ATA) and the other one has more short term lab environments for testing quick exploit scenarios and validating security ideas/scenarios.

The Hyper-V host running Windows 2016 TP4 hosts VMs with a Windows Server 2016 forest with Windows 2016 members servers and Windows 10 clients.

Lab Automation

I use customized PowerShell scripts to do most of the work for me. *I will post these scripts in my GitHub repository for use in the near future*; however, they should be considered example scripts and used only as a starting point for automating your environment.

Common Active Directory Troubleshooting Commands

DCDiag /c /v /e /fix /f:c:\DCDIAG.Log
Run a comprehensive test against all DCs in the forest with verbose logging

- /c: Performs a comprehensive suite of tests.
- /v: Provides verbose logging displaying additional information on what is being tested and the result.
- /fix: fixes any unregistered DC SPNs
- /a: Test all DCs in the site.
- /e: Tests ALL the DCs in the enterprise. Use with caution.
- /ReplSource:<SourceDomainController>: test connection between this DC and another.

http://technet.microsoft.com/en-us/library/cc731968%28v=ws.10%29.aspx

DCDiag Replication Related tests:

- CutOffServers
- Intersite
- MachineAccount
- NCSecDesc
- Netlogon
- ObjectsReplicated
- VerifyEnterpriseReferences
- VerifyRreplicas

NLTest /sc_query:DNSDomainName

Check Secure Channel

NLTest /sc_verify:DNSDomainName

Verifies Secure Channel

NLTest /dsgetsite

Check computer site (also checks secure channel)

Kllist -li 0x3e7

List Kerberos tickets for machine account

RPCDump

Shows replication rpc ports

Portqry

Run against port 135 to see mapped RPC ports.

Repadmin /SyncAll /A /e /P
Force a full forest replication synchronization of all partitions “pushing” changes out from the DC the command is run on.
http://technet.microsoft.com/en-us/library/cc770963%28v=ws.10%29.aspx

*Repadmin /options **

Check to see if any DC is misconfigured (Options)
http://technet.microsoft.com/en-us/library/cc736571%28v=ws.10%29.aspx#BKMK_38

Repadmin /replsummary

Forest wide replication health check

*Repadmin /kcc **

Forces KCC to run on all DCs

Repadmin /kcc /site:SITENAME

Forces KCC to run on all DCs in specified site

repadmin /removelingeringobjects ServerName ServerGUID DirectoryPartition /advisory_mode

http://technet.microsoft.com/en-us/library/cc785298(v=ws.10).aspx

Repadmin /bind

Check RPC connectivity

NOTE: If LinkValueReplication=NO, then it’s Windows 2000 Forest Functional Mode.
Repadmin /queue <DCNAME>

See replication queue
Or Perf counter: NTDS_DRA Pending Replication Synchronizations

Repadmin /showreps

Repadmin /showrepl /v

Information about replication partners – shows NEVER replicated DCs

Repadmin /showutdvec

Information about NC Up-to-dateness Vector

Repadmin /showconn

Information about connection objects

Repadmin /showsig

Shows InvocationID & Retired InvIDs

Repadmin /siteoptions SERVERNAME /site:SITENAME +Win2k3_Bridges_Required

When BASL is disabled, this site option configures Intersite Mesaging to develop the intersite cost matrix useful for DFS.
Repadmin /showobjmeta <ObjectDN>

See AD object history

- Legacy shows groups existing before Win2k3 Forest Funtional Level
- Present shows groups created/modified (group members removed/added) after Win2k3 Forest Funtional Level
- Recycle Bin deleted objects show here as Present but with DEL:GUID

Ipconfig /all

Ping ##.##.##.##

Nslookup ######.###

DNSCMD

DNSLint (KB 321045)

Active Directory Common Ports Used:

53	DNS
88	Kerberos
123	SNTP
135	RPC Endpoint Mapper
137	NetBIOS
138	NetBIOS
139	NetBIOS
389	LDAP
445	SMB

464	Kerberos Change Password
636	LDAP (SSL)
3268	Global Catalog
3269	Global Catalog (SSL)
5722	DFS-R (SYSVOL)
5985	WinRM
9389	ADWS (AD Powershell)

Windows Server 2008 (and newer) DCs use IANA RPC port range: 49152 – 65535

(Visited 129,805 times, 8 visits today)