* Part 1: 15 minutes
* Intro / Welcome
* Story of why / how this presentation came about
  + 2 years ago I went to a conference called Tech Mentor. (Shameless plug)
  + Attended all the DSC-related sessions I could – and returned determined to master it
  + Followed along with the Jeffrey Snover and Jason Helmick video on MVA
  + WMF 5.0 just released – with new requirements for certificates to encrypt credentials
    - Specific template for document encryption
  + Documentation (at the time) was non-existent
  + Configuring a template to do document encryption was difficult!
  + I wanted to make things easier
* Part 2 – build a PKI for an automated lab
  + Destroy and deploy – not a normal production use
  + Build ADCS
  + Group policy
  + Template
* Part 3 – DevOps and the Release Pipeline
  + Need this for automated build and test
* Part 4 – Continued evolution
  + I thought this was going to be easy – it’s just another config
  + I was wrong.
  + There were no resources to do any of this stuff – so I had to write it myself
  + This is what made it hard
  + Even Microsoft is interested in this solution
  + I’m going to show you how I solved the problem, yes there is code, and I will give it to you, but it’s not about the code.
  + It’s about understanding how to solve problems

Part 2: slides (15 minutes)

* Slide 1: Just a quick overview if you are new to DSC why you would need to encrypt credentials? (Ask audience)…
  + <c> LCM runs as localsystem
  + <c> give examples where privileges are not enough!
  + <c> privileged accounts NEED to be protected.
* Slide 2: Terminology and Acronyms
  + PKI – Public Key Infrastructure – can be external (think Verisign) or internal – this is what we’re talking about here
  + Root/Issuing/Policy CAs
  + AIA – accessible location to retrieve public certificates in the chain (LDAP, private or public website)
  + CDP – CDP, like an AIA, must be available to the clients using the certificates from the PKI
  + Standalone (non-AD-Integrated)
  + Enterprise (AD-integrated)
    - Makes management “easier”
* CA Tiering Options
  + One tier – cost-effective, but not very secure – root is online and private key vulnerable to compromise
  + Two-tier – protect the private key of the root by keeping it offline – drive should be protected
  + Three-tier – layer is added between the root and issuing
    - Policy CA allows restriction on types of certificates that can be issued from an issuing CA
    - Administrative boundary – not security boundary
    - Policy CAs can be kept offline also
  + Discuss – Standalone vs. Enterprise
* Best Practices
  + One tier not recommended for Production
  + Two vs. three depends on your requirements
* Challenges
  + Offline root
    - Definition of “offline”?
      * Doesn’t mean “put it on the network, configure it, shut it all down”
      * Means – should never touch the network
      * Means – No PowerShell remoting
      * Means – no “cross-node orchestration” in DSC – WaitFor items
      * Only method of getting a MOF File there is USB stick (and there are risks with that too)
  + Lack of PowerShell cmdlets
    - A few cmdlets for installation / configuring of the bits
    - A few for configuring CDP and AIA
    - Still a lot of things that can only be done using certutil / certreq – or write your own PowerShell if you’re really good with .Net
    - Cmdlets exist to create templates (add-CATemplate) but not configure them
    - Group Policy cmdlets (to set up auto-enrollment) also not great
  + Many moving parts – a 1-tier CA with DSC is easy, a 2-tier with an offline root is not
    - Need to publish CDP and AIA from a server that is online
    - Publish in LDAP, publish on website
    - Exchanging of files – issuing needs public cert of root, root needs to “issue” the issuing cert
  + Enterprise Issuing
    - Chicken vs. Egg with respect to credential encryption
    - Requires Enterprise Admin, but not on a domain controller.
    - Therefore, need \*credentials\*
    - If this is the first CA, there is no CA available to encrypt the credentials so need to be in plain text
    - Consider re-creating the MOF once you can encrypt the credentials!

Part 3: Show the code (15 mins)

* For this Demo
  + Two-tier, Standalone Root with an Enterprise Subordinate
  + ? Anyone in the audience built a CA with DSC before? How easy (or hard) did you find it?
  + Code – Show the interesting parts:
    - Note that there is a DC component to the config
    - 982 lines long – used all script resources – in future to be changed over to custom resources and added to Gallery
    - General orchestration:
      * Build “offline root”, share the certificate, wait.
      * DC: Publish root into AD, wait.
      * Subordinate: Wait for root certificate, wait for AD publish, import root, set up CRL distribution point in IIS, install ADCS. wait. Back to root.
      * Root: Get the cert request, issue the issuing, root completed.
      * Subordinate: Get issuing cert, install cert and start ADCS, configure CRLs and other settings, wait.
      * DC: GPO for Autoenrollment, create DSC template, end.
      * Subordinate: Publish DSC template, end.
    - WaitFor
    - IssueTheIssuing
      * The test portion checks to see if the certificate was already created.
        + 3-step process : certreq submit (already done when the subordinate is installed)
        + Certutil resubmit – to publish
        + Retrieve to write it out
        + Talk about why I did it this way
    - GPO (Create GPO, 3 registry settings, link the GPO) – show in main code
    - DSC Template – show in main code
    - Offline root:
      * Install ADCS (WindowsFeature) and Configure (xADCSDeployment)
      * Set CDP (script)
      * Clear AIA (script)
      * Set registry settings (Registry resource)
      * Publish CRL (Script)
      * Create a share with the public root certificate – for Enterprise (xSMBShare)
      * (Wait!)
    - Domain Controller
      * Create a DNS record for the CDP website
      * Copy public root certificate (File)
      * Publish root certificate to AD (script)
      * AutoEnrollment
      * DSC Template
    - Enterprise Issuing:
      * Wait for Root cert to be available and DSPublish
      * Copy public root certificate (File)
      * Import root to certificate store (script) – timing thing
      * Import the CRL (script)
      * Install ADCS and IIS (WindowsFeature)
      * Configure IIS for CRL distribution – mix of File, xWebAdministration, xSMBShare, and Script resources
      * Configure ADCS (xADCSDeployment)
    - Back to Root:
      * Grab Issuing certificate request from Issuing CA (file)
      * Issue the Issuing using certreq / Certutil / certreq
      * Copy stuff back to Issuing CA
        + Install Certificate
        + Set CRL/CDP/AIA
    - Autoenrollment
    - DSC Template: You need to add this to the code
      * Slide – DSC Template requirements – link to documentation
      * Show DSC Template in GUI and point out each one – Show Provider, under Extensions – Key Usage and Application Policies

Part 4: Demo the encryption

* + Demo: Encrypting credentials within a MOF using this template
    - Putting it all together: What do I need?
      * Need the public key to encrypt the credentials in the MOF.
      * Export the public key to a .cer file using export-PFXCertificate and copy it to the authoring box
      * Need to specify the .cer file in the config data for the config to be encrypted
      * Need to specify the thumbprint of the certificate on the LCM for decryption
      * And… boom 😊

Part 5: Questions?

* + Some final thoughts to ponder:
    - Is encrypting credentials overkill?
      * This was introduced as a solution when MOF files were \*not\* encrypted on the target (WMF 4.0)
      * Now, MOF files are encrypted on the target (WMF 5.0) – so credentials are too
      * Pull server – on-Prem vs. Azure Automation DSC
      * Authoring Box – Depends on the security of your authoring box
      * If it were me and DA credentials, I’d still be encrypting them!!