





# WSUS Monitoring and Testing

Thursday, March 26, 2015 10:01 AM

Problem: I need a way to test and verify computers are updated.

## Powershell Update Services API

 <http://blogs.technet.com/b/heyscriptingguy/archive/2013/04/02/windows-software-update-services.aspx>  
 <http://blogs.technet.com/b/heyscriptingguy/archive/2012/01/19/use-powershell-to-find-missing-updates-on-wsus-client-computers.aspx>  
 <https://msdn.microsoft.com/en-us/library/ms745830%28v=vs.85%29.aspx>  
 <https://msdn.microsoft.com/en-us/library/microsoft.update.services.administration%28v=vs.85%29.aspx>

```
[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration")
$Wsus = [Microsoft.UpdateServices.Administration.AdminProxy]::GetUpdateServer("SC1",$False,"8530")
$Wsus
$wsus.GetConfiguration()
$wsus.GetDatabaseConfiguration()
$wsus.GetUpdateEventHistory("$(Get-Date).AddHours(-1))", "$(Get-Date)")
```


### Limitations

From what I am looking at, this is looking at the WSUS server itself. I really want to look at the individual computer. Additionally, I want to be able to check against both the WSUS server on our LAN, as well as the Windows Update server.

### Try with COM Object

```
$updates = New-Object -ComObject "Microsoft.Update.UpdateColl"
```

## Resources:

	<a href="https://gist.github.com/jacobludriks/9ca9ce61de251a5476f1">https://gist.github.com/jacobludriks/9ca9ce61de251a5476f1</a>	wsus-update pshell
	<a href="https://msdn.microsoft.com/en-us/library/windows/desktop/aa387291%28v=vs.85%29.aspx">https://msdn.microsoft.com/en-us/library/windows/desktop/aa387291%28v=vs.85%29.aspx</a>	Microsoft.Update COM Object

## Technique: Changing Server Selection

```
$Criteria = "IsInstalled=0 and Type='Software'"
$Searcher = New-Object -ComObject Microsoft.Update.Searcher
$Searcher.ServerSelection
Shows 0, or default.
$Searcher.ServerSelection = 2
$Searcher.ServerSelection
shows 2 now.
```

### About

ServerSelection is an ENUM

```
typedef enum {
    ssDefault      = 0,
    ssManagedServer = 1,
    ssWindowsUpdate = 2,
    ssOthers        = 3
} ServerSelection;
```

From <[https://msdn.microsoft.com/en-us/library/windows/desktop/aa387280\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa387280(v=vs.85).aspx)>

### Testing:

It worked. Setting ServerSelection to 2, and running the scan works!

## Function: get-windowsupdateversion

```
function get-windowsupdateversion{
    $WUVer = (New-Object -com "Microsoft.Update.AgentInfo")
```

```
$WUObj = New-Object -TypeName System.Object
Add-Member -InputObject $WUObj -MemberType NoteProperty -Name "Product Version" -Value $WUVer.GetInfo("ProductVersionString")
$WUObj
}
```

From <[https://msdn.microsoft.com/en-us/library/windows/desktop/aa385815\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa385815(v=vs.85).aspx)>

## A Working Scan against Windows Update Servers

```
[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration")
$Criteria = "IsInstalled=0 and Type='Software'"
$Searcher = New-Object -ComObject Microsoft.Update.Searcher
$Searcher.ServerSelection
$Searcher.ServerSelection = 2
$SearchResult = $Searcher.Search($Criteria).Updates
$SearchResult
```

## Function: Getting the KBArticleIDs out of \$SearchResult

```
$SearchResult | Select KBArticleIDs | foreach {$_.KBArticleIDs}
```

## ITTeam6902dg Evaluation

```
Title : Update for Microsoft Camera Codec Pack for Windows 8.1 for x64-based Systems (KB2899189)
Title : Windows 8.1 Update for x64-based Systems (KB2919355)
Title : Update for Windows 8.1 for x64-based Systems (KB2938066)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2966826)
Title : Update for Windows 8.1 for x64-based Systems (KB2967917)
Title : Security Update for Microsoft .NET Framework 4.5.1 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2894856)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2973114)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2972213)
Title : Security Update for Microsoft .NET Framework 4.5.1 and 4.5.2 on Windows 8.1 and Windows Server 2012 R2
x64-based Systems (KB2977765)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2894852)
Title : Update for Windows 8.1 for x64-based Systems (KB2990967)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2966828)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2968296)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2972103)
Title : Security Update for Microsoft .NET Framework 4.5.1 and 4.5.2 on Windows 8.1 and Windows Server 2012 R2
x64-based Systems (KB2978041)
Title : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for x64-based
Systems (KB2979573)
Title : Security Update for Microsoft .NET Framework 4.5.1 and 4.5.2 on Windows 8.1 and Windows Server 2012 R2
x64-based Systems (KB2979576)
Title : Update for Windows 8.1 for x64-based Systems (KB2994290)
Title : Update for Windows 8.1 for x64-based Systems (KB3012199)
Title : Update for Windows 8.1 for x64-based Systems (KB3008242)
Title : Update for Windows 8.1 for x64-based Systems (KB3013816)
Title : Update for Windows 8.1 for x64-based Systems (KB2989930)
Title : Update for Windows 8.1 for x64-based Systems (KB3013769)
Title : Update for Windows 8.1 for x64-based Systems (KB3013410)
Title : Update for Windows 8.1 for x64-based Systems (KB3004394)
Title : Microsoft .NET Framework 4.5.2 for Windows 8.1 and Windows Server 2012 R2 for x64-based Systems (KB2934520)
Title : Security Update for Windows 8.1 for x64-based Systems (KB3023266)
Title : Security Update for Windows 8.1 for x64-based Systems (KB3004365)
Title : Security Update for Windows 8.1 for x64-based Systems (KB3019978)
Title : Security Update for Windows 8.1 for x64-based Systems (KB3022777)
```

Title : Security Update for Windows 8.1 for x64-based Systems (KB3019215)  
 Title : Update for Windows 8.1 for x64-based Systems (KB3031044)  
 Title : Update for Windows 8.1 for x64-based Systems (KB3016074)  
 Title : Update for Windows 8.1 for x64-based Systems (KB3020338)  
 Title : Security Update for Windows 8.1 for x64-based Systems (KB3023562)  
 Title : Security Update for Windows 8.1 for x64-based Systems (KB3004361)  
 Title : Security Update for Windows 8.1 for x64-based Systems (KB3000483)  
 Title : Update for Windows 8.1 for x64-based Systems (KB3000850)  
 Title : Update for Windows 8.1 for x64-based Systems (KB3048778)  
 Title : Update for Windows 8.1 for x64-based Systems (KB2976978)

## Finding the Applied Updates from PowerShell

SCCM Connect. (sccm command in my \$profile)

\$SUG = "March Insanity"

#(Get-CMSoftwareUpdateGroup ).LocalizedDisplayName #Displays the Groups we have.

(Get-CMSoftwareUpdate -UpdateGroupName \$SUG).LocalizedDisplayName

Finding Bulletin IDs

(Get-CMSoftwareUpdate -UpdateGroupName \$SUG).BulletinID

Finding Article IDs (KB Articles)

(Get-CMSoftwareUpdate -UpdateGroupName \$SUG).ArticleID

## Getting KB Article IDs for Software Update Groups

sccm

\$sug = "March Insanity"

\$insanity = (Get-CMSoftwareUpdate -UpdateGroupName \$SUG).ArticleID

\$sug = "March Madness"

\$madness = (Get-CMSoftwareUpdate -UpdateGroupName \$SUG).ArticleID

\$sug = "Patch Tuesday 2015-03-26 16:27:47"

\$tuesday = (Get-CMSoftwareUpdate -UpdateGroupName \$SUG).ArticleID

## Getting KB Article IDs for Local Machine

[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration")

\$Criteria = "IsInstalled=0 and Type='Software'"

\$Searcher = New-Object -ComObject Microsoft.Update.Searcher

\$Searcher.ServerSelection = 2

\$SearchResult = \$Searcher.Search(\$Criteria).Updates

\$locals = \$SearchResult | Select KBArticleIDs | foreach {\$\_.KBArticleIDs}

## Comparing the Results

### Variables

\$insanity	Article IDs for Insanity Push
\$madness	Article IDs for Madness Push
\$tuesday	Article IDs for Patch Tuesday Push
\$locals	Missing Article IDs for Local Computer

compare-object \$locals \$tuesday -ExcludeDifferent -IncludeEqual

## Identifying Local Severity

\$severity = \$SearchResult | Select MsrcSeverity | foreach {\$\_.MsrcSeverity}

\$searchResult | Select MsrcSeverity, Title

\$SearchResult

\$severity = \$SearchResult | Select MsrcSeverity | foreach {\$\_.MsrcSeverity}

\$severity

\$searchResult | Select MsrcSeverity

```
$searchResult | Select MsrcSeverity, Title
$searchResult | Where-Object {$_.MsrcSeverity -eq "Critical"}
$searchResult | Where-Object {$_.MsrcSeverity -eq "Critical"} | Select MsrcSeverity, Title | fl
```

```
Title      : Windows 8.1 Update for x64-based Systems (KB2919355)
Title      : Security Update for Microsoft .NET Framework 3.5 on Windows 8.1 and Windows Server 2012 R2 for
             x64-based Systems (KB2972103)
Title      : Security Update for Microsoft .NET Framework 4.5.1 and 4.5.2 on Windows 8.1 and Windows Server 2012 R2
             x64-based Systems (KB2978041)
Title      : Security Update for Windows 8.1 for x64-based Systems (KB3000483)
```

## More Comparisons

```
$insanity.Contains("2919355")
```

## Scratchpad

```
psexec -s \\$computer c:\windows\system32\winrm.cmd quickconfig -quiet
```

```
$computername_crit = ScanRemoteCriticals -Computers "computername" -Credential $cred
$computername = ScanRemoteArticles -Computers "computername" -Credential $cred
```

# [ps] Exchange Receive Connector

Tuesday, May 30, 2017 11:57 AM

I needed the user SC1\_ScheduledTasks to be able to send emails as Service Desk to any user.

Since SC1\_ScheduledTasks did not have a mailbox in exchange

\$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri <http://exchcas/PowerShell/>

Import-PSSession \$Session -AllowClobber -DisableNameChecking

Add-ADPermission -Identity "Service Desk" -user SC1\_ScheduledTasks -AccessRights ExtendedRight -ExtendedRights "Send As"

Add-ADPermission -Identity "IP Secured Inbound" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-SMTP-Submit

Add-ADPermission -Identity "Default Frontend EXCHCAS1" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-SMTP-Submit

Add-ADPermission -Identity "Default Frontend EXCHCAS2" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-SMTP-Submit

Add-ADPermission -Identity "Default Frontend EXCHCAS1" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-SMTP-Accept-Any-Recipient

Add-ADPermission -Identity "Default Frontend EXCHCAS1" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-Bypass-Message-Size-Limit

Add-ADPermission -Identity "Default Frontend EXCHCAS1" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-Bypass-Anti-Spam

Add-ADPermission -Identity "Default Frontend EXCHCAS2" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-SMTP-Accept-Any-Recipient

Add-ADPermission -Identity "Default Frontend EXCHCAS2" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-Bypass-Message-Size-Limit

Add-ADPermission -Identity "Default Frontend EXCHCAS2" -user SC1\_ScheduledTasks -ExtendedRights ms-Exch-Bypass-Anti-Spam

## References

Extended Permissions: [https://technet.microsoft.com/en-us/library/jj673053\(v=exchg.150\).aspx](https://technet.microsoft.com/en-us/library/jj673053(v=exchg.150).aspx)

# Siggy Change Log

Friday, June 12, 2015 9:22 AM

## Testing 1.70

Connected computer to domain.

Moved computer into Affected OU for targeting Signature.

### First Logon:

- ! Registry Data Type Haze
- ! Payload didn't drop - seemed to get stuck in waiting for outlook
- ! Hard to debug, since data isn't written to log immediately.

### Second Logon

Registry data looks correct.

Payload didn't drop. Still stuck on outlook

## Modifications 1.71

- ★ Added Local-Log Function
- ★ Added Check-LocalLogging Function

## Testing 1.71 ITS6902DG

- ! Remote Log only written if payload executes.
- 💡 Reason - EndProgram is never called when Program works.

## Modifications 1.72

- 💡 Added condition for clean sig to call EndProgram.
- 💡 Added startup info to Local-Log for \$siggyVersion and \$username.

## Testing 1.72 ITVM0612AA

- ! Payload launched, but glitched. IT failed to copy the signature over.
- 💡 Reason - \$signaturesourcefiles was undefined

## Modifications 1.73

- 💡 Added \$sigsource to the Copy Item in verify signature.

## Modifications 1.74

- 💡 Moved the Copy-Item up in Check-SignatureSync. This was failing because the function would always return, and never actually copy the signature
- 💡 Added a 3 minute sleep timer to main function. This allows for the environment to have some time to "bake" before we even begin to run. I have a feeling that being patient may take care of some issues for us.

## Testing 1.74

### Windows 7, 32 bit

- ★ Deleted Signature from Roaming profile, ran siggy. it worked.

## Modifications 1.75

- 💡 Changed Signature to Old harbin Signature.

- 💡 Deployed to Family Practice OU for testing.

## Testing 1.75

- ★ After changing signature, verified signature will adjust in citrix, and on windows 7 DGs
- ! AD is synchronized to Registry BEFORE the payload is delivered. If the PAYLOAD is never delivered on the first run, there will be no second run. Critical Flaw.
- ★ This even works in Outlook within Xenapp

## Modifications 1.77

- 💡 Split CheckADSync into 2 functions. Check-ADSync and Synchronize-AD. Check-AD is read only, Synchronize-AD will be used ONLY after the payload is verified to have run.

## Testing 1.77

- ! Replacement strings were wrong if Fax, or Cell number were missing on the template.
- ! Old Harbin signature didn't have the updated replacement token Harbmail, and was still set to Email. Outlook Wait works great. It has worked on a user that took 2 hours and 4 minutes to start outlook. It waited patiently

## Modifications 1.78

- 💡 Fixed replacement strings for template (harbmail)
- 💡 Fixed Null Cell or Office number to remove the correct line of text.
- 💡 Added a feature to replace phone numbers with a dotted notation whenever there is a minus sign
- 💡 Fixed an issue where \$username was not displayed in both the local and remote log.
- 💡 Siggy Write-Log had unused \$stamp variable from old version. Cleaned.
- 💡 Added Timestamp to Write-Log instead of Finalize Log. This gives every item in the Remote Log its own timestamp.

## Modifications 1.78.2

- 💡 Removed Duplicate Timestamp from Finalize Log.
- 💡 Changed wording in the Log function on Check-ADSync. It had previously stated that it was updating values, but this is a read only function.
- 💡 Fixed an issue on the Fax number that did not replace the minus signs in the number with dots.
- 💡 Removed all references to \$loginfo and \$NL.
- 💡 Changed old-harbin signature from Mobile MobileNumber to Cell MobileNumber. This will mean that if no cell is added, the reference to the number will be correctly removed by the script
- 💡 Fully Linked GPO to FAMPRAC and added a few other smaller departments.

## Testing 1.78.2

- ! Fax number isn't replacing dotted

## Modifications 1.78.4

- 💡 Fixed Fax Number in code.
- 💡 Added a -force switch to the script for easier testing. This disables the 3 minute timer before evaluations, and bypasses outlook wait.

## Testing 1.78.4

- ! Users complain that their credentials (CMA) etc have not been added to the end of their title.

## Modifications 1.78.7

- 💡 Added Doctor Logic. Since Doctors AD fields are different.
- 💡 Added a Comment Field. This pulls from AD Attributes (Comment)



Changed signature to Harbin\_Standard.docx

## Testing 1.78.7

We experienced LOTS of feedback on 1.78.7. I sent out a mass email, giving each user a picture of what their signature would look like after it was generated.



# Is anybody using Chrome in contosoville

Monday, July 25, 2016 3:24 PM

```
$searchbase =
```

```
$users = get-aduser -SearchBase "OU=Domain Users,DC=Contoso,DC=Com" -filter *
```

```
$sessions = get-xasession
```

```
$arr = @()
```

```
foreach ($user in $users){
```

```
    $arr += "clinic\"+$user.samaccountname
```

```
}
```

```
foreach ($session in $sessions){
```

```
    #Write-Host $session.accountName -noNewLine
```

```
    Write-Host $session.accountName
```

```
    if($arr -contains $session.AccountName){
```

```
        $vals = $session | Select ServerName,AccountName,BrowserName,ClientName
```

```
    }
```

```
    else {
```

```
        #Write-Host "no."
```

```
    }
```

```
}
```

# [ps] Calculated Properties

Thursday, June 1, 2017 9:46 AM

## The Setup: Create an Object

\$object is set to a value of the first 5 processes returned by Get-Process

```
$object = (Get-Process)[0..4]
```

## Iteration 1: Basic Syntax

```
$object | Select-Object Name , @{  
    Name="Home";  
    Expression={"Simple String"}  
}
```

## Iteration 2: Using Properties in the Pipe

```
$object | Select-Object Name , @{  
    Name="Home";  
    Expression={$_.path}  
}
```

## Iteration 3: Using a Script Block

Since Expression is literally just an anonymous scriptblock, we can define a scriptblock somewhere else in the script, and then reference it inside the Hash of Select-Object. It would look like this

```
$block = {  
    $random = Get-Random -Minimum 1 -Maximum 10  
    $name = $_.Name  
    $Ret = "$name$random"  
    return $ret  
}  
$object | Select-Object Name , @{  
    Name="Home";  
    Expression=$block  
}
```

# [ps]Add Reservation

Wednesday, October 5, 2016 9:51 AM

```
Get-DhcpServerv4Scope -ComputerName contoso-dc3
Get-DhcpServerv4Lease -ScopeId 172.16.0.0 -ComputerName contoso-dc3
Get-DhcpServerv4Lease -ScopeId 172.16.0.0 -ComputerName contoso-dc3 | out-gridview
```

Add the reservation

```
Add-DhcpServerv4Reservation -ScopeId 172.16.0.0 -ComputerName contoso-dc3 -IPAddress 172.16.75.150 -ClientId
"000000000001"
```

## Adding DNS Records

First, you need to determine your zone.

```
Get-DnsServerZone -ComputerName contoso-dc3
```

```
Add-DnsServerResourceRecordA -Name storefront.contoso.com -IPv4Address 172.16.75.150 -ComputerName contoso-dc3 -
ZoneName contoso.com
```

# [ps]MostRecent Reservations in Each Scope

Monday, August 29, 2016 11:43 AM

## For Single Scope

```
$scopeIDs = Get-DhcpServv4Scope -ComputerName "contoso-dc3" | Select ScopeID
```

```
Get-DhcpServv4Lease -ComputerName contoso-dc3 -scopeID $scopeIDs[0].ScopeID.IPAddressToString
```

```
Get-DhcpServv4Lease -ComputerName contoso-dc3 -scopeID $scopeIDs[1].ScopeID.IPAddressToString
```

## Sort by Active, then sort by date

```
Get-DhcpServv4Lease -ComputerName contoso-dc3 -scopeID $scopeIDs[1].ScopeID.IPAddressToString  
| ? {$_.AddressState -eq "Active"} | Sort LeaseExpiryTime -ascending
```

## Select 5 most recent

```
Get-DhcpServv4Lease -ComputerName contoso-dc3 -scopeID $scopeIDs[1].ScopeID.IPAddressToString  
| ? {$_.AddressState -eq "Active"} | Sort LeaseExpiryTime -Descending | select -First 5
```

## contoso

```
function test(){  
$adair = Get-DhcpServv4Lease -ScopeID 192.168.0.0 -computername contoso-dc3 | ?  
{$_.AddressState -eq "Active"} | Sort LeaseExpiryTime -Descending | Select -First 5  
  
$scopes = Get-DHCPservv4Scope -ComputerName contoso-DC3 | ? {$_.state -eq "Active"}  
foreach ($scope in $scopes){  
Write-Host -foreground yellow ("testing " + $scope.Name)  
$list = Get-DhcpServv4Lease -ScopeID $scope.ScopeID -computername contoso-dc3 | ?  
{$_.AddressState -eq "Active"} | Sort LeaseExpiryTime -Descending | Select -First 5  
foreach ($item in $list){  
Write-Host -Foreground Green ("testing " + $item.IPAddress + " " + $item.HostName)  
Test-Connection -computername $item.ipaddress -count 1  
}  
}  
}  
while($true){  
test  
}
```

## [PS][AD] Right Click Properties

Thursday, May 21, 2015 9:21 AM

### ADSI, AdminContextMenu

go.ps1

```
Function Clean-LDAP{
    param([string]$LDAPQuery)

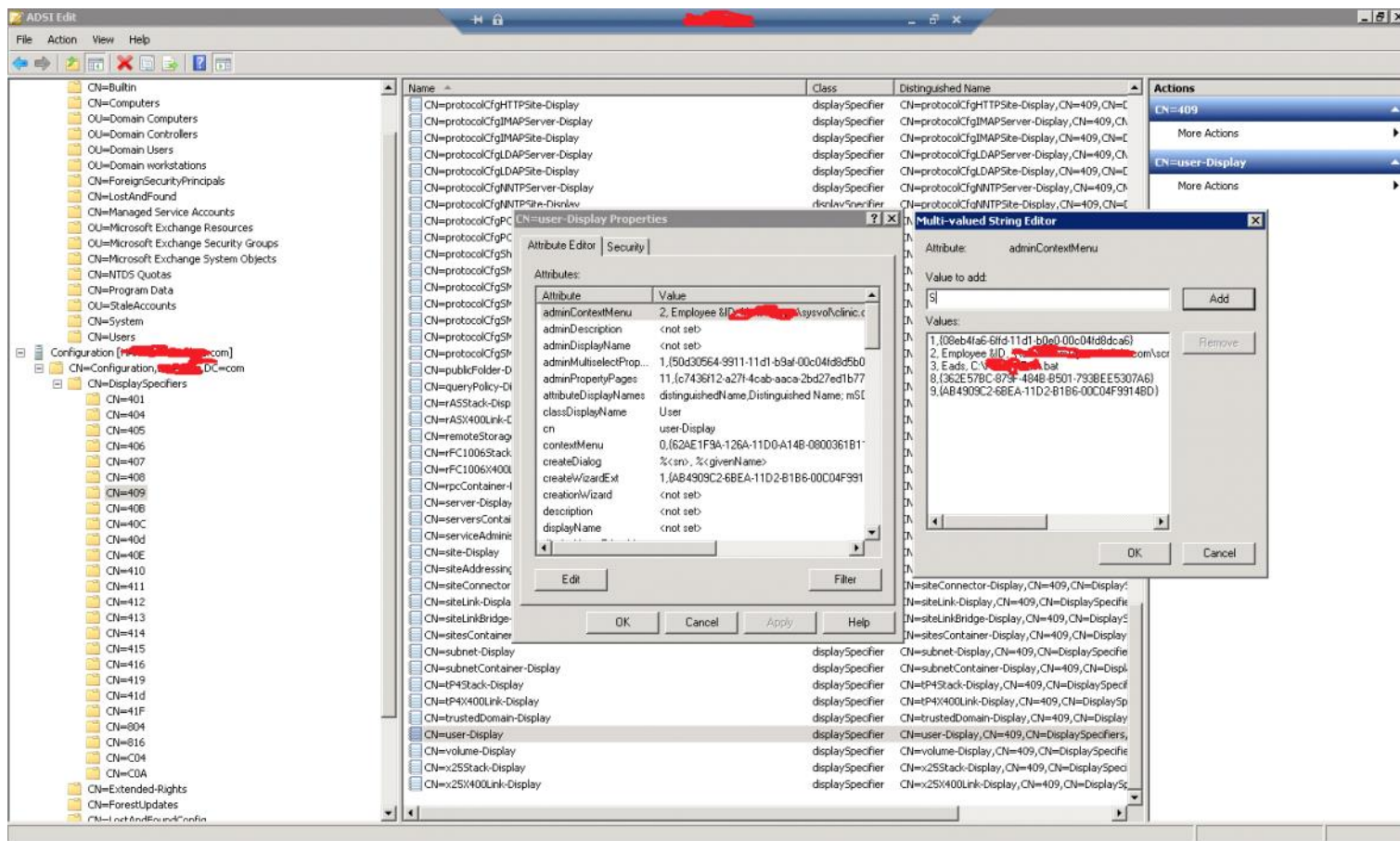
    $remove = "^LDAP://contoso-DC\d.contoso.com/"
    return ($LDAPQuery -replace $remove, "")
}
Function Main{
    param([string]$LDAPQuery)
    $Search = Clean-LDAP -LDAPQuery $LDAPQuery
    Write-Host "Final Search is:`t$search"
    Get-ADUser $Search
}
}
```

test.bat

```
@echo off
echo %1
powershell -NoProfile ". C:\contoso\go.ps1; Main -LDAPQuery "%1""
pause
```

### Admin Context Menu Value

3, Eads, C:\Harbin\Test.bat



# Diagnosing Account Lockouts

Wednesday, April 29, 2015 2:08 PM

## Step 1: Get the Scan

### Run this

```
cd .\programming\toolkit
.. \lockouts.ps1
ScanForLockouts -hours 48
```

### Next Steps

After running those commands, powershell will scan each of our Domain Controllers for lockouts. All of the lockout related events will be stored in \$output in powershell.

### How the scan works

The ScanForLockouts function takes the optional parameter -hours. It will scan each of the Domain Controllers Security Event log going for events within the past \$hours. 48 is a good number to use for this one.

It is specifically looking for events with Event ID of 4740 or 4767

<https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4767>

### After the scan runs

you can export all of the objects to your clipboard by using

```
$output | format-list | clip
```

Paste the events into notepad, from there, you can save the output, or just skim the document to find more information.

### Limit to Specific User

\$output will be all account lockout events. To simplify search, lets limit it to the specific user

```
$cleaned = $output | Where-Object {$_.Message -match "fred.flinstone"}
```

Now send cleaned output to clipboard. Things should be much easier to read.

```
$cleaned | Format-List | clip
```

### Find the Cause of the Lockout

Having the lockout events from ScanForLockouts essentially only gives you the times that an event occurred. It does not give you the information on what actors have caused the lockout.

From your output, look at the times generated, and search for those events specifically.

## The Time Based Scan

```
$Occurrence = New-Object DateTime
```

```
$Occurrence = [DateTime]"4/30/2015 8:51:00am"
```

# Netscaler Work

Friday, February 26, 2016 9:55 AM

From <<https://secureportal.citrix.com/Licensing/Downloads/RetrieveLicense.aspx?PRID=8682>>

From <<https://secureportal.citrix.com/Licensing/Downloads/RetrieveLicense.aspx?PRID=8682>>

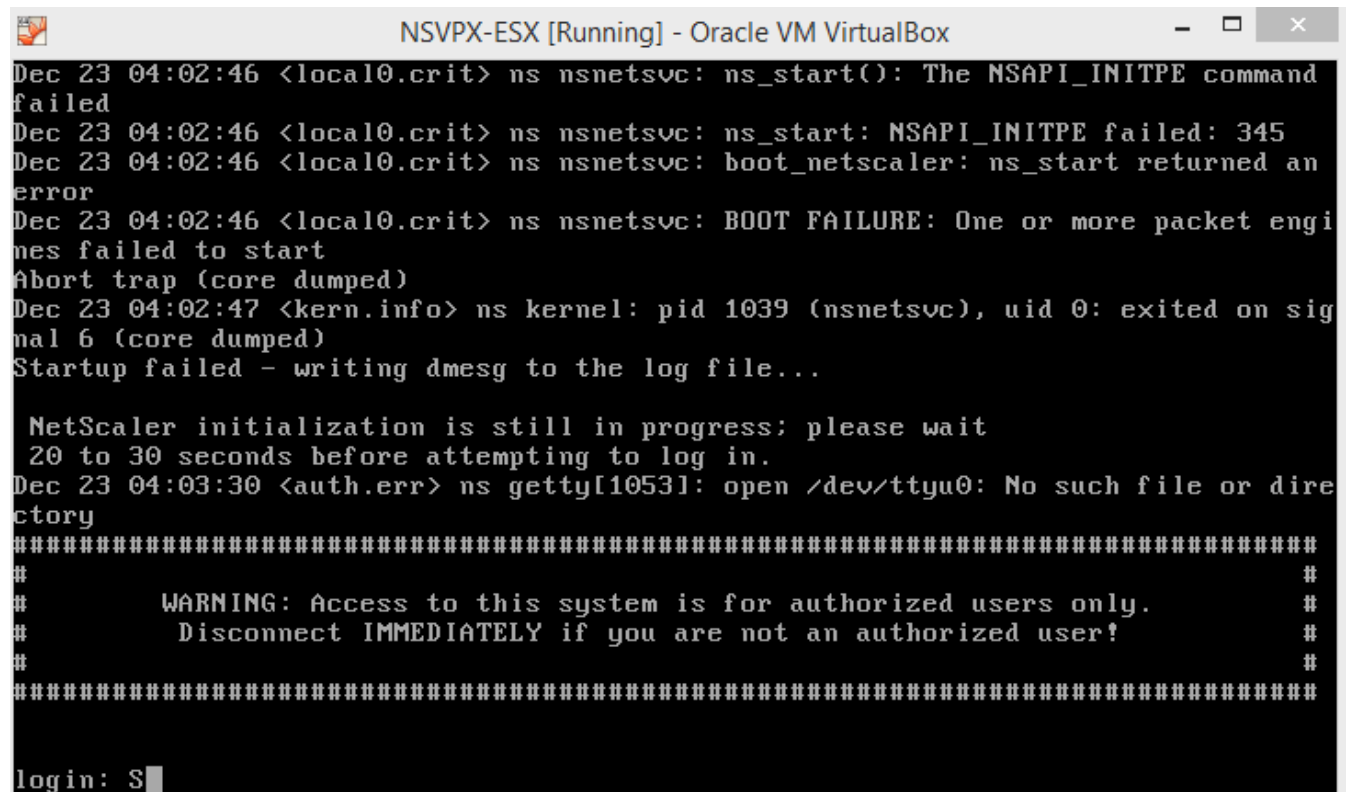
## Straddling the DMZ Process

[http://support.citrix.com/article/CTX136926?\\_ga=1.11584125.898696167.1455597081](http://support.citrix.com/article/CTX136926?_ga=1.11584125.898696167.1455597081)

In its default configuration, a NetScaler appliance does not provide any differentiation between network interfaces. All network traffic is transmitted on all interfaces. The appliance functions more like a network hub than a switch. This can lead to Layer 3 network loops where duplicated traffic is transmitted on multiple interfaces.

- ☐ Verify that this is the configuration scheme that we are using

## NSVPX-ESX



```
NSVPX-ESX [Running] - Oracle VM VirtualBox
Dec 23 04:02:46 <local0.crit> ns nsnetshvc: ns_start(): The NSAPI_INITPE command
failed
Dec 23 04:02:46 <local0.crit> ns nsnetshvc: ns_start: NSAPI_INITPE failed: 345
Dec 23 04:02:46 <local0.crit> ns nsnetshvc: boot_netscaler: ns_start returned an
error
Dec 23 04:02:46 <local0.crit> ns nsnetshvc: BOOT FAILURE: One or more packet engi
nes failed to start
Abort trap (core dumped)
Dec 23 04:02:47 <kern.info> ns kernel: pid 1039 (nsnetshvc), uid 0: exited on sig
nal 6 (core dumped)
Startup failed - writing dmesg to the log file...

NetScaler initialization is still in progress; please wait
20 to 30 seconds before attempting to log in.
Dec 23 04:03:30 <auth.err> ns getty[1053]: open /dev/ttyu0: No such file or dire
ctory
#####
#
#      WARNING: Access to this system is for authorized users only.      #
#      Disconnect IMMEDIATELY if you are not an authorized user!          #
#
#####
login: S
```

Try to trick it into thinking its running on ESX instead of Virtual Box

VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVendor"

"Phoenix Technologies LTD"

VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVersion" "6.00"

VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseDate"

"07/31/2013"

VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMajor" 6

VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMinor" 0

```
VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMajor"
6
VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMinor"
0
VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiSystemVendor"
"VMware, Inc."
VBoxManage setextradata NSVPX-ESX "VBoxInternal/Devices/pcbios/0/Config/DmiSystemProduct"
"VMware Virtual Platform"
```

## Setup the IP on the virtual

```
set ns config -ipaddress 10.90.34.101 -netmask 255.255.0.0
add ns ip 10.50.24.200 255.255.0.0 -type snip
add route 0.0.0.0 0.0.0.0 10.90.100.10
```

```
save ns config
reboot
```

```
add dns nameserver 192.168.1.1
add dns nameserver redacted
```

Source:

<https://jayceechou.wordpress.com/2009/01/20/netcaler-basic-configuration/>

## Primary Netscaler

**10.90.34.101** - 255.255.0.0 - Gateway: 10.90.100.10

## Configure and Understand SNIP

SNIPs are also known as interface IPs. Every interface/VLAN you plug into the NetScaler needs a SNIP associated with that interface/VLAN.

**SNIP = 10.90.34.200**

## Lets Get the licenses

```
root@NSTest1# Imutil Imhostid -ether
```

Imutil - Copyright (c) 1989-2013 Flexera Software LLC. All Rights Reserved.

The FlexNet host ID of this machine is "0800277aed15"



[Home](#) / [My Account](#) / All Licensing Tools

[Overview](#) [View Licenses](#) [Activate and Allocate Licenses](#) [Single Allocation](#) [Reallocate](#) [Redownload](#) [Return](#)

Results per page 
Page  of 1


<input checked="" type="checkbox"/>	Host ID	Host ID Type	License File
<input checked="" type="checkbox"/>	0800277aed15	MAC Address	<a href="#">See details</a>

Results per page 
Page  of 1

## Licenses

License type	Platinum	Model ID	1
Load Balancing	✓	SSL Offloading	✓
Content Switching	✓	Cache Redirection	✓
Global Server Load Balancing	✓	GSLB Proximity	✓
Authentication, Authorization and Auditing	✓	NetScaler Gateway	✓
Maximum NetScaler Gateway Users Allowed	5	Maximum ICA Users Allowed	Unlimited
Clustering	✓	Web Interface	✓
Integrated Caching	✓	Front End Optimization	✓
Rewrite	✓	Responder	✓
HTTP Compression	✓	Content Filtering	✓
Application Firewall	✓	Cloud Bridge	✓
Priority Queuing	✓	Sure Connect	✓
Surge Protection	✓	DoS Protection	✓
AppFlow	✓	AppFlow for ICA	✓
IPv6 Protocol Translation	✓	Dynamic Routing	✓
BGP Routing	✓	OSPF Routing	✓
RIP Routing	✓	ISIS Routing	✓
Content Accelerator	✓	AppQoS	✓
NetScaler Push	✓	Web Logging	✓
vPath	✓	RISE	✓
Callhome	✗	Large Scale NAT	✓

## Setup Access2.contoso

 VIP **192.168.1.101**

## Check configs of NS1 192.168.2.101 (working production)

... Redacted ...

## Monitor Change - Force Failover tonight

I ran this on 192.168.redacted and this may fix the portal issue  
set vpn parameter -UITHEME DEFAULT