Windows Server Update Services

Windows Server Update Services (WSUS) est un service permettant de distribuer les mises à jour pour Windows et d'autres applications Microsoft sur les différents ordinateurs fonctionnant sous Windows au sein d'un parc informatique. WSUS est un rôle pour serveur Windows lui permettant ainsi de devenir un serveur de mises à jour local (ou proxy de mises à jour). Ce serveur télécharge et stocke ponctuellement l'ensemble des mises à jour disponibles auprès des serveurs Windows Update de Microsoft et rend possible le contrôle de la diffusion de celles-ci dans le parc.

Outils autour de WSUS

Wsus Package Publisher permet de publier vos propres mises à jour sous forme de fichiers MSI, MSP ou EXE. Ainsi, vous pouvez déployer des applications telles qu'Adobe Reader, Java, Flash Player ou Symantec Endpoint Protection. Et les mettre à jour.

Vous pourrez importer des mises à jour à partir de catalogues d'éditeur (Adobe, Dell, HP...). Afin de publier des pilotes ou firmware pour votre matériels.

Official git repo : https://github.com/DCourtel/Wsus_Package_Publisher
Tutoriel Vidéo : PlayList de Francis Bonnamour

WAPT

WAPT est un logiciel de déploiement son noyau est sous licence GPLv3.

Site Officiel : https://wapt.fr/
Documentation : https://wapt.fr/en/doc/
Official git repo : https://github.com/tranquilit/WAPT

WSUS Offline

WSUS Offline Update est un logiciel de gestion des mises à jour destiné à certaines applications et systèmes d'exploitation de Microsoft. Contrairement au service WSUS de Microsoft, WSUS Offline Update permet d'enregistrer les mises à jour sur un média pour les distribuer vers différents postes d'un parc informatique hors-ligne (offline) au réseau Internet.
Site Officiel : https://www.wsusoffline.net/

Script

Ici, je vais positionné certains scripts que j'ai trouvé pour gagner de la place.

Function Invoke-WsusDeclineAllSuperSeded

```
function Invoke-WsusDeclineAllSuperSeded
{
    param(
        [Parameter(HelpMessage='Set the fully qualified domain name like
this : myhost.example.com. The FQDN uniquely distinguishes the device from
any other hosts called myhost in other domains.')]
        [String]$FQDN,
        [Parameter(HelpMessage='If your Wsus use a SSL certificate, please
set this parameter to $true, by default this parameter is set on $false.')]
        [Boolean]$useSecureConnection = $false,
        [Parameter(HelpMessage='Set the communication port, by default this
parameter is set on 8530')]
        [Int32]$portNumber = 8530
    )

    if(!$FQDN){
        [String]$FQDN = $env:COMPUTERNAME + $(if (($null -eq
$env:USERDNSDOMAIN) -eq $false){'.' + $env:USERDNSDOMAIN})
    }

# Load .NET assembly

[void][reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.A
dministration")

[Int32]$count = 0

# Connect to WSUS Server

Try {
    $updateServer =
[Microsoft.UpdateServices.Administration.AdminProxy]::getUpdateServer($FQDN,
$useSecureConnection, $portNumber)
    write-output "<<<Connected sucessfully >>>"
}
Catch {
    Write-Warning "An error occurred:"
    Write-Error $_
    Break
}



    $updateServer.GetUpdates($(New-Object
Microsoft.UpdateServices.Administration.UpdateScope)) | ForEach-Object {
        if ($_.IsSuperseded -eq 'True') {
            Write-Output ("Decline Update : $_.Title")
```

```
            $_.Decline()
            $count = $count + 1
        }
    }
    Total Declined Updates: $count

trap {
    Write-Warning -Message 'Error Occurred'
    Write-Warning -Message 'Exception Message: '
    Write-Error $_.Exception.Message
    Write-Error $_.Exception.StackTrace
    exit
}
}
# Please Set your settings here
Invoke-WsusDeclineAllSuperSeded
# EOF
```