SCRIPT #15 · ACTIVE DIRECTORY

Reset password in blocco con notifica email

Reimposta la password per un elenco di utenti e invia le nuove credenziali via email.

AUTORE	COPYRIGHT	PREREQUISITI
Francesco Larossa	© 2026 SysAdmin Italia	Modulo ActiveDirectory, server SMTP raggiungibile

Cosa fa questo script

Lo script legge un elenco di utenti da un CSV, genera per ciascuno una password casuale sicura, la imposta come nuova password con obbligo di cambio al primo accesso, e invia automaticamente una email di notifica con le credenziali temporanee.

Come si usa

Configura prima $SmtpServer e $MittenteEmail. Esegui in simulazione, verifica le password generate mostrate a console, poi ripeti con -Simula:$false per applicare realmente le modifiche.

```
.\Reset-PasswordInBlocco.ps1 -PercorsoCSV "C:\Dati\utenti.csv"
# dopo verifica:
.\Reset-PasswordInBlocco.ps1 -PercorsoCSV "C:\Dati\utenti.csv" -Simula:$false
```

In quale contesto usarlo

Scenario tipico: un evento di sicurezza che richiede il reset forzato di un gruppo di account, oppure la preparazione di credenziali per un gruppo di nuovi utenti prima del primo giorno lavorativo. Da usare con attenzione, dato che trasmette credenziali via email in chiaro.

Codice completo

Il codice è lo stesso del file Reset-PasswordInBlocco.ps1: per usarlo scarica il file dalla pagina invece di copiarlo da questo PDF.

```
<#
.SYNOPSIS
    Reimposta la password per un elenco di utenti (da CSV) e invia una notifica email.
.NOTES
    Richiede un server SMTP configurato.
#>
param(
    [Parameter(Mandatory=$true)]
    [string]$PercorsoCSV,
    [string]$SmtpServer = "smtp.azienda.local",
    [string]$MittenteEmail = "it-support@azienda.local",
    [switch]$Simula = $true
)

Import-Module ActiveDirectory

$utenti = Import-Csv -Path $PercorsoCSV -Delimiter ";"
```

```
function New-PasswordCasuale {
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 12 | ForEach-Object {[char]$_}) + "!1"
}

foreach ($u in $utenti) {
    $nuovaPassword = New-PasswordCasuale

    if ($Simula) {
        Write-Host "SIMULAZIONE: reset password per '$($u.Username)', nuova password: $nuovaPassword" -ForegroundColor Cyan
        continue
    }

    try {
        Set-ADAccountPassword -Identity $u.Username -NewPassword (ConvertTo-SecureString $nuovaPassword -AsPlainText -Force) `
            -Reset
        Set-ADUser -Identity $u.Username -ChangePasswordAtLogon $true

        $corpo = "Ciao,`n`nLa tua password è stata reimpostata. Nuova password temporanea: $nuovaPassword`n`nIT Support"
        Send-MailMessage -From $MittenteEmail -To $u.Email -Subject "Reset password account" -Body $corpo `
            -SmtpServer $SmtpServer

        Write-Host "OK: password reimpostata e notificata per '$($u.Username)'" -ForegroundColor Green
    }
    catch {
        Write-Host "ERRORE per '$($u.Username)': $_" -ForegroundColor Red
    }
}
```