# Classe de schéma vulnérable

ID METSYS : 13  
ID PingCastle : [S-ADRegistrationSchema](https://pingcastle.com/PingCastleFiles/ad_hc_rules_list.html)  
ID ANSSI : [vuln2\_warning\_schema\_posssuperiors](https://www.cert.ssi.gouv.fr/uploads/guide-ad.html#warning_schema_posssuperiors)

Charge de travail : 1 jour  
Difficulté : 3 - Requière des connaissances avancées sur l'anomalie

****Description****

Les classes ajoutées au schéma fournissent des types d'objets supplémentaires. Si elles sont mal configurées, une classe peut être utilisée pour contourner une restriction de sécurité.

• Pour la vulnérabilité « PossSuperiorComputer » : Une classe possède l'attribut « possSuperiors » contenant la classe « computer », et cette classe hérite de « container ». Cela signifie que chaque ordinateur peut demander l'ajout de cette classe. Une fois cette classe ajoutée, elle peut être utilisée comme conteneur pour créer des utilisateurs ou des ordinateurs supplémentaires sans restriction.

• Pour la vulnérabilité « PossSuperiorUser » : Il s'agit de la même vulnérabilité que « PossSuperiorComputer », mais avec la classe « user » au lieu de la classe « computer ».

****Recommandation****

• Pour « PossSuperiorComputer » : Vous devez modifier le schéma pour changer la valeur de l'attribut « possSuperior » et supprimer la valeur « computer ».

• Pour « PossSuperiorUser » : Vous devez modifier le schéma pour changer la valeur de l'attribut « possSuperior » et supprimer la valeur « user ».

De plus, il est connu que la classe « msExchStorageGroup » présente cette vulnérabilité via la CVE-2021-34470. Dans ce cas, la vulnérabilité est exploitable même si Exchange a été désinstallé.

Script PowerShell pour trouver les schémas vulnérables :

# Dictionary to hold superclass names

$superClass = @{}

# List to hold class names that inherit from container and are allowed to live under computer object

$vulnerableSchemas = [System.Collections.Generic.List[string]]::new()

# Resolve schema naming context

$schemaNC = (Get-ADRootDSE).schemaNamingContext

# Enumerate all class schemas

$classSchemas = Get-ADObject -LDAPFilter '(objectClass=classSchema)' -SearchBase $schemaNC -Properties lDAPDisplayName,subClassOf,possSuperiors

# Enumerate all class schemas that computer is allowed to contain

$computerInferiors = $classSchemas |Where-Object possSuperiors -eq 'computer'

# Populate superclass table

$classSchemas |ForEach-Object {

$superClass[$\_.lDAPDisplayName] = $\_.subClassOf

}

# Resolve class inheritance for computer inferiors

$computerInferiors |ForEach-Object {

$class = $cursor = $\_.lDAPDisplayName

while($superClass[$cursor] -notin 'top'){

if($superClass[$cursor] -eq 'container'){

$vulnerableSchemas.Add($class)

break

}

$cursor = $superClass[$cursor]

}

}

# Outpupt list of vulnerable class schemas

$vulnerableSchemas