Question
Scott Roth · Jun 29, 2018

Adding TLS to ZAUTHENTICATE

I wrote a ZAUTHENTICATE.mac a couple of months back, and found recently that it is creating coredumps on almost a nightly basis. I think I have figured out this problem to be not clearing out my MsgSearch after I am doing 2 of them within the code.

1. Get User Attibutes from AD

2. Get User Groups From AD

So while I am trying to cleanup the code I thought it would be a good time to add a Certificate and TLS to the mix since I should of been using that all along. However I keep running into issues

Error message: Cache error: <UNDEFINED>ZAUTHENTICATE+104^ZAUTHENTICATE *LD

its not displaying the error code it should be from the ZAUTHENTICATE in the Audit Database. How do I get it to tell me where it is actually stopping in the ZAUTHENTICATE code? Or can someone look at the code below and see what I might be doing wrong?

```
ZAUTHENTICATE(ServiceName,Namespace,Username,Password,Credentials,Properties) PUBLIC {
#include %occErrors
#include %sySecurity
#include %sySite
#include %syLDAP
#define LDAPServer $Get(^OSUMCLDAP("Server"))
#define WindowsLDAPServer 1
#define WindowsCacheClient 0
#define UseSecureConnection 1
#define UnixCertificateFile $Get(^OSUMCLDAP("LDAPKey"))_"certnew.pem"
#define WindowsBaseDN "dc="_$Get(^OSUMCLDAP("Domain"))_",dc=edu"
#define WindowsFilter "sAMAccountname"
#define WindowsAttributeList $lb("displayName","department","mail")
 s $zt="Error"
 s Status = 0
 i Password="" {
s Status= $SYSTEM.Status.Error($$$InvalidUsernameOrPassword)
g Error
 }
 i $$$WindowsLDAPServer{
s AdminDN=$Get(^OSUMCLDAP("User"))
s AdminPW=$Get(^OSUMCLDAP("Pass"))
 }
 i $$$ISWINDOWS,$$$UseSecureConnection{
 s LD=##Class(%SYS.LDAP).Init($$$LDAPServer)
 i LD=0 {
 s Status=##Class(%SYS.LDAP).GetLastError()
 s Status="Init error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status)
 g Error
  }
 s Status=##Class(%SYS.LDAP).SetOption(LD,$$$LDAPOPTXTLSCACERTFILE,$$$UnixCertificateFile)
```

```
 i Status'=$$$LDAPSUCCESS{
 s Status ="SetOption error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status)
 g Error
}
  s Status=##class(%SYS.LDAP).StartTLSs(LD)
  i Status'=$$$LDAPSUCCESS{
  s Status="ldap_setoption(Certificate) - "_##class(%SYS.LDAP).Err2String(Status)
  g Error
  }
 }

 s Status=##Class(%SYS.LDAP).SimpleBinds(LD,AdminDN,AdminPW)
 i Status'=$$$LDAPSUCCESS
  {
s Status = "ldap_Simple_Bind(AdminDN) - "_##Class(%SYS.LDAP).Err2String(Status)
#;w !,Status
g Error
  }
  i $$$WindowsLDAPServer {
  s Filter=$$$WindowsFilter_"="_Username
 }
 i $$$WindowsLDAPServer {
s AttributeList=$$$WindowsAttributeList
#;AttributeList
 }
 i $$$WindowsLDAPServer {
s BaseDN=$$$WindowsBaseDN
#;BaseDN
 }
  s SearchScope=$$$LDAPSCOPESUBTREE
  s Timeout=30
  s SizeLimit=1
 s Status=##Class(%SYS.LDAP).SearchExts(LD,BaseDN,SearchScope,Filter,AttributeList,0,"","",Timeout,"",.
SearchResult)
 i Status'=$$$LDAPSUCCESS {
i Status=$$$XLDAPFILTERERROR {
s Status="1,User "_Username_" does not exist"
w !,Status
} else {
s Status=Status_",ldap_Search_Ext - "_##Class(%SYS.LDAP).Err2String(Status)
}
g Error
 }
 s NumEntries=##Class(%SYS.LDAP).CountEntries(LD,SearchResult)
 i NumEntries=-1 {
 s Status=##Class(%SYS.LDAP).GetError(LD)
 s Status=Status_",ldap_Count_Entries - "_##Class(%SYS.LDAP).Err2String(Status)
 g Error
 }
 i NumEntries=0 {
s Status="1,User "_Username_" does not exist"
  g Error
 }
 i NumEntries>1 {
s Status="1,LDAP Filter is not unique"
  g Error
 }
 s CurrentEntry=##Class(%SYS.LDAP).FirstEntry(LD,SearchResult)
 i CurrentEntry=0 {
```

```
 s Status=##Class(%SYS.LDAP).GetError(LD)
 s Status="ldap_FirstEntry - "_##Class(%SYS.LDAP).Err2String(Status)
 g Error
 }
 s DN=##Class(%SYS.LDAP).GetDN(LD,CurrentEntry)
 i Password="" {
 s Status="1,ldap_Simple_Bind("_DN_") - password cannot be null"
 g Error
 }
 s Status=##Class(%SYS.LDAP).SimpleBinds(LD,DN,Password)
 i Status'=$$$LDAPSUCCESS {
 s Status=Status_",ldap_Simple_Bind("_DN_") - "_##Class(%SYS.LDAP).Err2String(Status)
 g Error
 }
 s Attribute=##Class(%SYS.LDAP).FirstAttribute(LD,CurrentEntry,.Ptr)
 while (Attribute'="") {
   s Values=##Class(%SYS.LDAP).GetValuesLen(LD,CurrentEntry,Attribute)
   #;Values:"_Values
   s Properties("Attributes",Attribute)=Values
  s Attribute=##Class(%SYS.LDAP).NextAttribute(LD,CurrentEntry,.Ptr)
 }
 s Properties("Username")=Username
 s Properties("FullName")=$li(Properties("Attributes","displayName"))
 k Properties("Attributes","displayName")
 s Properties("Comment")=$li(Properties("Attributes","department"))
 k Properties("Attributes","department")
 s Properties("EmailAddress")=$li(Properties("Attributes","mail"))
 k Properties("Attributes","mail")
 i $d(SearchResult) d ##Class(%SYS.LDAP).MsgFree(SearchResult)
 s GroupFilter="(&(objectClass=group)(member:1.2.840.113556.1.4.1941:="_DN_"))"
 s GroupAttributes=""
 s Status=##Class(%SYS.LDAP).SearchExts(LD,BaseDN,$$$LDAPSCOPESUBTREE,GroupFilter,
GroupAttributes,0,"","",10,0,.GroupSearchResult)
 i Status'=$$$LDAPSUCCESS {
w !,"SearchExts error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status)
g Error
 }
 s GroupNumEntries=##Class(%SYS.LDAP).CountEntries(LD,GroupSearchResult)
 i GroupNumEntries=-1 {
s Status=##Class(%SYS.LDAP).GetError(LD)
s Status=##Class(%SYS.LDAP).Err2String(Status)
g Error
 }
 w !
 i GroupNumEntries=0 {
w !,"No nested groups for "_Username_" found"
g Done
 }
 i GroupNumEntries>0 {
//w !,"Found "_GroupNumEntries_" nested groups for user "_Username
 }
 s GroupCurrentEntry=##Class(%SYS.LDAP).FirstEntry(LD,GroupSearchResult)
 i GroupCurrentEntry=0 {
s Status=##Class(%SYS.LDAP).GetError(LD)
w !,"FirstEntry error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status)
g Error
 }
 s Groups=""
 While (GroupCurrentEntry'=0) {
```

```
s GroupDN=##Class(%SYS.LDAP).GetDN(LD,GroupCurrentEntry)
i GroupDN="" {
s Status=##Class(%SYS.LDAP).GetError(LD)
w !,"GetDN Group error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status)
g Error
}
s CN=$p(GroupDN,",",1)
s AD=$p(CN,"=",2)
s AD=$zcvt(AD,"L")
s exists="$d(^|"%SYS"|SYS("Security","RolesD",AD))
i exists{
s Properties("Roles") = AD
}
  s GroupCurrentEntry=##Class(%SYS.LDAP).NextEntry(LD,GroupCurrentEntry)
 }
Done
 //i $d(SearchResult) d ##Class(%SYS.LDAP).MsgFree(SearchResult)
 i $d(GroupSearchResult) d ##Class(%SYS.LDAP).MsgFree(GroupSearchResult)
#;Close the connection and free the LDAP in memory structures.
 i +$d(LD) d ##Class(%SYS.LDAP).UnBinds(LD)
 #;w !,"SystemOK "_$SYSTEM.Status.OK()
 q $SYSTEM.Status.OK()
Error s $zt=""
 i $d(SearchResult) d ##Class(%SYS.LDAP).MsgFree(SearchResult)
 i $d(GroupSearchResult) d ##Class(%SYS.LDAP).MsgFree(GroupSearchResult)
 i +$d(LD) s Status=##class(%SYS.LDAP).UnBinds(LD)
 i $ze'=""{
 #;w !,"ERROR:"_$SYSTEM.Status.Error($$$CacheError,$ze)
 q $SYSTEM.Status.Error($$$CacheError,$ze)
 } else{
  #;w !,"ERROR:"_$SYSTEM.Status.Error($$$GeneralError,"LDAP error: "_Status_" -
"_##Class(%SYS.LDAP).Err2String(Status))
  q $SYSTEM.Status.Error($$$GeneralError,"LDAP error: "_Status_" - "_##Class(%SYS.LDAP).Err2String(Status))
 }
}
```

Thanks

Scott Roth

The Ohio State University Wexner Medical Center

#LDAP #Object Data Model #Security #SSL #Caché

Source URL: https://community.intersystems.com/post/adding-tls-zauthenticate