.net column

Role-Based security vanuit .NET

Voor de loskoppeling van gebruikers en te autoriseren applicatieonderdelen kan gebruik gemaakt worden van .NET role-based security. Het voordeel van .NET role-based security ten opzichte van andere implementaties, zoals COM+, is dat het de mogelijkheid biedt om zowel gebruik te maken van Windows accounts als van zelf gedefinieerde applicatieaccounts.

Role-based security maak niet alleen functionele autorisatie mogelijk, maar biedt tevens de mogelijkheid om op gegevensniveau te controleren in welke rol de gebruiker zit. Als voorbeeld, stel dat de service "*Overboeken*" een operatie "*MaakGeldOver*" kent. Bij functionele autorisatie wordt er gecontroleerd of de gebruiker lid is van één van de applicatierollen, die toegekend is aan de service "Overboeken". Zo zullen alle medewerkers in de rollen "Baliemedewerkers" en "Toezichthouders", functioneel gebruik moeten kunnen maken van de "Overboeken" service. De controle op gegevensniveau maakt het mogelijk om onderscheid te maken tussen "Baliemedewerkers" en "Toezichthouders". Zo mogen baliemedewerkers bijvoorbeeld tot maximaal 500.000 euro overmaken en toezichthouders hebben geen limiet.

Role-based security in .NET werkt op basis van zogenaamde *principal*- en *identity*-objecten, die beschikbaar zijn op de uitvoerende thread van de applicatie.

Het Identity-object encapsuleert informatie over de gebruiker, zoals de naam en het authenticatietype. Het .NET framework maakt onderscheid tussen een *GenericIdentity*- en een *WindowsIdentity*-object. Het *GenericIdentity*-object kan gebruikt worden in een custom logon scenario, waarbij de gebruikers geen Windows account hebben. Het *WindowsIdentity*-object wordt toegepast indien de applicatie gebruikt maakt van Windows authenticatie.

Het *Principal*-object vertegenwoordigt de security-context waaronder code wordt uitgevoerd en bevat een lijst met rollen. Tevens heeft het principal object een verwijzing naar een identity object, waarmee de gebruiker wordt geïdentificeerd.

Het .NET framework kent twee soorten Principals te weten een *GenericPrincipal*- en een *WindowsPrincipal*- object. Het *GenericPrincipal*- object biedt de mogelijkheid om eigen rollen te verzinnen, terwijl de rollen van het *WindowsPrincipal*-object overeen komen met Windows groepen.

In het volgende voorbeeld wordt getoond hoe er vanuit code een nieuwe principal en identity aan de huidige thread kan worden toegekend:

```
GenericIdentity  myIdentity
= new
GenericIdentity("beheerder");
String [] Roles = new
String[]
{"Baliemedewerkers",
"Toezichthouders"};
GenericPrincipal myPrincipal
= new
GenericPrincipal(myIdentity,
Roles);
Thread.CurrentPrincipal =
myPrincipal;
```

De vraag is echter, hoe controleren we nu of de gebruiker in de juiste rol zit die benodigd is om gebruik te kunnen maken van een bepaalde service?

Het .NET framework biedt een zogenaamd *PrincipalPermission*-object waarmee de Common Language Runtime autorisatiecontroles kan uitvoeren. Het *PrincipalPermission*-object controleert de rol of de identiteit van de aanroepende thread. Het *PrincipalPermission*-object kan zowel declaratief als imperatief (vanuit code) worden gebruikt.

In het volgende voorbeeld wordt getoond hoe role-based security declaratief kan worden gebruikt

```
[PrincipalPermission(Security
Action.Demand,

Role="Baliemedewerkers")]
[PrincipalPermission(Security
Action.Demand,

Role="Toezichthouders")]
public static void
MaakGeldOver(...)
{
if(Thread.CurrentPrincipal.Is
InRole("Baliemedewerkers"))
{
  // Mag maximaal tot
500.000 overmaken.
  // Dit is een voorbeeld
van gegevensautorisatie
}
}
```

Het .NET framework biedt een role-based security mechanisme dat flexibel en uitbreidbaar genoeg is om aan de autorisatie behoefte van vele applicaties te kunnen voldoen. Het mooie van .NET role-based security is dat het ruimte biedt voor zowel Windows gebruikers (WindowsIdentity) als niet Windows gebruikers (GenericIdentity). Het PrincipalPermission-object maakt het mogelijk om te controleren of de gebruiker over de juiste rol beschikt.

Xander Buffart is IT Architect bij Info Support