Authentication and authorization strategies in ASP.NET applications

Cătălin MAICAN
Universitatea *Transilvania* din Braşov

Authentication and authorization strategies consist of choosing what resources will be protected, by means of what techniques, also taking into account some considerations regarding the users, the types of platforms from where the applications would be accessed, and the transition of identities along tiers.
Keywords: *authentication and authorization strategy, role-based authentication, choosing an authorization mechanism.*

Introducere

Crearea unei strategii de autentificare şi autorizare pentru aplicaţii Web distribuite nu este o sarcină oarecare. Totuşi, prin răspunsul următoarele întrebări se poate lua o decizie în acest sens:
- unde se va utiliza autorizarea şi ce mecanisme se vor utiliza?
- ce mecanisme de autentificare se vor utiliza?
- se va utiliza *Active Directory* pentru autentificare sau se va utiliza un depozit personalizat?
- care sunt consideraţiile şi implicaţiile de design pentru platforme eterogene şi omogene?
- cum vor fi reprezentaţi utilizatorii care nu utilizează sistemul de operare Windows?
- care va fi cursul procesului de identitate de-a lungul nivelelor (*tier*) aplicaţiei; când se utilizează impersonarea sau delegarea oferită de sistemul de operare?

În momentul utilizării autorizării, trebuie luat în calcul şi procesul de autentificare, deoarece orice politică de autorizare necesită utilizatori autentificaţi şi modalitatea de autentificare a utilizatorilor determină *porţile de acces* necesare aplicaţiei.

Alegerea unei strategii

Prin următorii paşi se poate identifica un proces care ajută în dezvoltarea unei strategii de autentificare şi autorizate pentru o aplicaţie:
1. identificarea resurselor;
2. alegerea unei strategii de autorizare;
3. alegerea identităţilor pentru accesul la resurse;
4. considerarea fluxului identităţilor;
5. alegerea unei modalităţi de autentificare;
6. alegerea modalităţii de transmitere a identităţii.

Identificarea resurselor presupune alegerea acelor resurse pe care aplicaţia trebuie să le expună clienţilor. Între resursele tipice se numără:
a. resurse ale server-lor Web (pagini Web, servicii Web), precum şi resurse statice (pagini HTML şi imagini);
b. resurse din baza de date, precum date disponibile numai pentru utilizatori sau date disponibile în întreaga aplicaţie;
c. resurse de reţea, precum sisteme de fişiere la distanţă şi datele din depozite de tip *Active Directory*;

Tot în această etapă trebuie identificate şi resursele pe care trebuie să le acceseze aplicaţia, acestea deosebindu-se de resursele expuse clienţilor. Printre acestea se numără: *Registry*, jurnalele de evenimente, fişierele de configuraţie.

Alegerea unei strategii de autorizare presupune două strategii de bază:
- bazată pe roluri – accesul la operaţii (metode) este securizat pe baza apartenenţei la roluri a apelantului. Rolurile sunt utilizate pentru baza de utilizatori a aplicaţiei în mulţimi de utilizatori care partajează aceleaşi privilegii de securitate în aplicaţie. Utilizatorii sunt mapaţi la roluri, iar dacă utilizatorul este autorizat să execute operaţia cerută, aplicaţia utilizează identităţi fixe cu care să acceseze resursele. Aceste identităţi sunt *de încredere* pentru managerii de resurse (baze de date, sisteme de fişiere etc.);

- bazată pe resurse – resursele individuale sunt securizate prin ACL din Windows. Aplicația va impersona apelantul înainte de a face verificările standard. Toate accesurile la resurse sunt executate prin contextul de securitate original al apelantului. Această impersonare are un impact negativ asupra scalabilității aplicației deoarece nu pot fi utilizate plajele de conexiuni în aplicație.

În marea majoritate a aplicațiilor Web, în care scalabilitatea este esențială, se recomandă autorizarea bazată pe roluri, în celelalte cazuri putând fi utilizată şi strategia de autorizare pe baza resurselor.

Modelul potrivit pentru autorizarea bazată pe roluri este:

- autentificarea utilizatorilor în aplicația Web front-end;
- maparea utilizatorilor la roluri;
- autorizarea accesului la operații (şi nu la resurse) pe baza apartenenței la roluri;
- accesarea resurselor back-end necesare prin utilizarea de identități fixe.

Alegerea identităților pentru accesul la resurse se face prin aflarea răspunsului la întrebarea *cine va accesa resursele*, în felul acesta putându-se alege identitatea sau identitățile care ar trebui să acceseze resursele de-a lungul nivelelor aplicației. Resursele pot fi accesate din aplicații Web, servicii Web, Enterprise Services şi .NET Remoting.

În toate aceste cazuri, identitatea utilizatorilor pentru accesul la resurse poate să fie:

- *identitatea originală a apelantului* – acest lucru presupune un model de impersonare/delegare în care identitatea originală poate fi obținută şi apoi transferată prin toate nivelurile aplicației. Factorul *delegație* este criteriul cheie utilizat pentru a determina mecanismul de autentificare;
- *identitatea procesului* – este cazul implicit (fără impersonare specifică). Apelurile la resursele locale sunt făcute prin utilizarea identității procesului curent. Fezabilitatea acestei operații depinde de granițele de trecut, deoarece identitatea procesului trebuie să fie recunoscută de sistemul țintă. Acest lucru implică faptul că apelurile sunt făcute astfel:
  - o în acelaşi domeniu de securitate Windows;
  - o de-a lungul mai multor domenii de securitate (utilizând conturi de încredere, sau nume de utilizatori şi parole duplicate în cazul în care nu există relații de încredere);
- *contul de serviciu* – această modalitate utilizează un cont de serviciu (fix). De exemplu, accesul la o bază de date poate fi făcut prin utilizarea unui nume de utilizator şi a unei parole fixe;
- *identitatea personalizată* – în cazul în care nu există conturi Windows de lucru, se poate construi propria identitatea (utilizând implementările IPrincipal şi IIdentity) care conțin detalii despre propriul context de securitate (liste de roluri, identificatori unici sau orice alte tipuri de informații).

Considerarea fluxului identităților este necesară pentru autorizarea pentru fiecare utilizator în parte, audit sau regăsirea de date personalizate, aceste lucruri fiind posibile prin transmiterea identității originale a apelantului prin diverse nivele ale aplicației şi prin diverse granițe între calculatoare.

Pe baza necesităților de autorizare la nivel de manager de resurse cât şi a necesităților de audit, trebuie identificate identitățile care trebuie să fie transmise prin aplicație.

Alegerea unei modalități de autentificare este influențată atât de natura utilizatorilor aplicației (tipul de browsere şi existența conturilor Windows) cât şi de necesitățile aplicației referitoare la impersonare/delegare şi audit.

Alegerea modalității de transmitere a identității presupune fie transmiterea identității (pentru a oferi un context de securitate) la nivel de aplicație, fie transmiterea acesteia şi a contextului de securitate la nivel de sistem de operare.

Pentru a transmite identitatea la nivel de aplicație se pot utiliza parametrii pentru proceduri stocate sau pentru metode. Transmiterea identității la nivel de sistem de operare presupune un model de impersonare/delegare.

Autorizarea bazată pe roluri pentru aplicațiile Web

Cele mai multe aplicații Web vor utiliza o autorizare bazată pe roluri. În acest caz trebuie luate în considerare mai multe tipuri de roluri, alegându-se dintre acestea tipul cel mai potrivit pentru scenariul în cauză. Există ast-

fel următoarele opţiuni: roluri .NET; roluri Enterprise Services; roluri definite pentru baza de date SQL Server; roluri pentru aplicaţie definite în baza de date.

Rolurile .NET sunt extrem de flexibile şi evoluează în jurul obiectelor de tip *IPrincipal* care conţine o listă de roluri care aparţin unei identităţi autentificate. Aceste roluri pot fi utilizate împreună cu aplicaţiile Web, serviciile Web sau cu componente găzduite de ASP.NET şi accesate utilizând HttpChanel.

Autorizarea se poate face fie utilizând aceste roluri în mod declarativ, prin folosirea *PrincipalPermission*, fie prin cod, utilizând cereri PrincipalPermission sau metoda IPrincipal.IsInRole în mod imperativ.

În cazul în care aplicaţia utilizează autentificarea de tip Windows, ASP.NET construieşte în mod automat un obiect *WindowsPrincipal* care este ataşat contextului de securitate al cererii curente (folosind HpptContext.User). După finalizarea procesului de autentificare şi ataşare a obiectului la cerere, acesta va fi utilizat pentru toate cererile care folosesc autorizarea bazată pe roluri .NET. Apartenenţa la grupurile Windows pentru apelanţii autentificaţi este utilizată pentru a determina mulţimea de roluri, în acest caz grupurile fiind considerate roluri.

Dacă aplicaţia utilizează un mecanism de autentificare în afara Windows, precum Forms sau Passport, trebuie creat propriul obiect GenericPrincipal (sau un obiect personal IPrincipal) care se va popula cu mulţimea de roluri existente într-o bază de date pentru utilizatorul curent.

Acest mecanism de securitate bazat pe roluri este şi extensibil, putându-se crea propriile clase care să implementeze IPrincipal şi IIdentity, extinzând în acest fel funcţionalitatea de bază. Astfel, atât timp cât în obiectul personal IPrincipal există roluri obţinute dintr-un depozit oarecare şi acesta este ataşat cererii curente, există o funcţionalitate de bază. De asemenea, se pot implementa şi metode şi proprietăţi care să extindă funcţionalitatea – de exemplu, crearea unei metode *IsInMultipleRoles (string[] roles)*, care va permite testarea apartenenţei la mai multe roluri.

Rolurile Enterprise Services împing verificarea accesului către middle-tier, permiţând utilizarea plajelor de conexiuni pentru bazele de date back-end. Cu toate acestea, pentru o autorizare bazată pe rolurile Enterprise Services, aplicaţia web front-end trebuie să impersoneze şi să transmită identitatea originală a apelantului (utilizând un jeton Windows) către aplicaţia Enterprise Services, acest lucru obţinându-se prin setări în Web.config:

```
<authentication mode="Windows" />
<identity impersonate="true" />
```

În cazul în care este suficientă verificarea declarativă a rolurilor la nivel de metodă pentru a determina ce utilizatori pot apela metoda respectivă, apartenenţa la roluri poate fi modificată prin instrumentul Component Services.

Rolurile definite pentru baza de date SQL Server se creează roluri în baza de date, se asignează permisiile pe baza acestora, urmând ca grupurile Windows şi conturile utilizatorilor să fie mapate rolurilor. În acest fel, identitatea utilizatorului trebuie transmisă către back-end.

Roluri pentru aplicaţie definite în baza de date – permisiile sunt acordate rolurilor dintr-o bază de date, dar rolurile nu conţin utilizatori sau grupuri de conturi. Dezavantajul este pierderea granularităţii apelantului original.

Prin utilizarea rolurilor pentru aplicaţii se poate face autorizarea pentru acces specific la anumite aplicaţii. Aplicaţia activează rolul utilizând o procedură internă care primeşte ca parametrii un nume de utilizator şi o parolă. Acest tip de autorizare are şi dezavantaje, şi anume faptul că aplicaţia trebuie să gestioneze datele utilizatorilor în mod securizat.

Alegerea unui mecanism de autentificare

În vederea alegerii unui mecanism eficient de autentificare trebuie luate în considerare următoarele elemente:

- *identităţile* – un mecanism de autentificare bazat pe Windows este potrivit numai în cazul în care utilizatorii au conturi Windows care pot fi autentificate printr-o autoritate de certificare accesibilă serverului Web care execută aplicaţia;

- *gestiunea datelor utilizatorilor* – unul din avantajele autentificării Windows este acela că permite sistemului de operare să gestioneze datele utilizatorilor. Prin utilizarea unei autentificări care nu se bazează pe Windows, trebuie luat în considerare şi depozitul pentru stocarea acestor date. Cele mai frecvente depozite sunt:
  - o baze de date SQL;
  - o obiecte-utilizatori din Active Directory.
- *transmiterea identității* – în cazul în care există necesitatea implementării unui mecanism de impersonare/delegare şi de transmitere a identității originale de-a lungul componentelor, aplicația trebuie să fie capabilă să impersoneze apelantul şi să delege contextul să de securitate către un subsistem;
- *tipul browser-ului* – în tabelul1 sunt ilustrate modalitățile de autentificare suportate de browser-ul Internet Explorer, precum şi de alte browsere.

Tabelul 1. Modalități de autentificare suportate de browser-e

Tip de autentificare	Necesită Internet Explorer	Observații
Forms	Nu	-
Passport	Nu	-
Integrată în Windows	Da (Kerberos şi NTLM)	Kerberos necesită Windows 2000/.NET atât pe clienți cât şi per servere.
Basic	Nu	Este parte a protocolului HTTP 1.1 care este suportat de toate browser-ele.
Digest	Da	
Certificate	Nu	Clienții necesită certificare X.509.

Pentru un scenariu cu aplicație care necesită acces la Internet, presupunerile sunt următoarele:
- utilizatorii nu dețin conturi Windows în domeniul serverului sau într-un domeniu de încredere accesibil serverului;
- utilizatorii nu dețin certificate la nivel de client.

În acest caz, se poate utiliza arborele de decizie din figura 1 pentru a se alege cel mai potrivit mecanism de autentificare. Între avantajele autentificării de tip Forms se numără:
- suport pentru autentificarea printr-un depozit oarecare de date (bază de date SQL Server sau Active Directory);
- suport pentru autorizarea bazată pe roluri, împreună cu căutarea rolurilor în baza de date;
- integrarea uşoară cu interfața Web pentru utilizator;
- ASP.NET oferă cea mai mare parte din infrastructura necesară, fiind necesar relativ puțin cod față de ASP clasic.

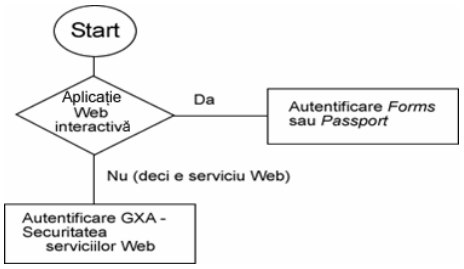

Fig. 1. Arbore de decizie pentru scenariu de tip Internet

Intre avantajele autentificări de tip Passport se numără:
- Passport este o soluție centralizată (dar este controlată de o autoritate unică, ceea ce poate constitui şi un dezavantaj);
- Elimină gestiunea datelor utilizatorilor din aplicație;
- Poate fi utilizat cu scheme de autorizare pe bază de roluri;
- Este securizat, fiind bazat pe tehnologii de criptare.

Pentru un scenariu Intranet/Extranet, în vederea luării deciziei potrivite în privința mecanismului de autentificare, se poate lua în considerare arborele de decizie din Fig.

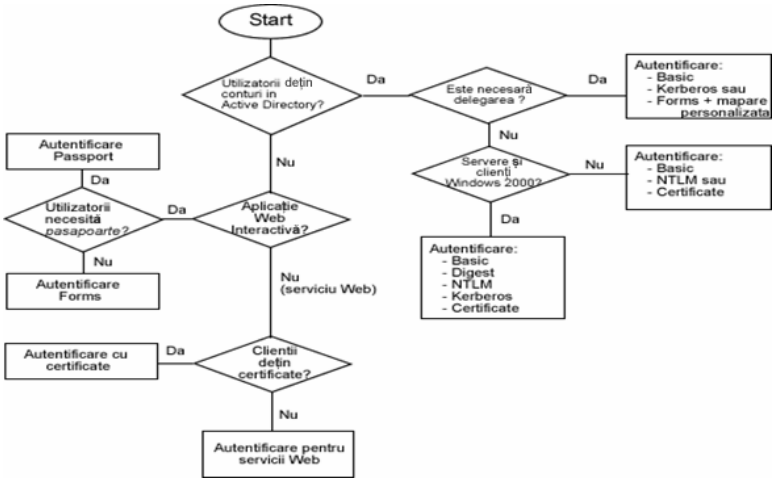

Fig. 2. Scenariu intranet/extranet

În Tabelul 2 este prezentată o scurtă comparație a mecanismelor de autentificare.

Tabelul 2. Tipuri de autentificare

	Basic	Digest	NTLM	Kerberos	Certificate	Forms	Passport
Utilizatorii au nevoie de conturi în domeniul serverului?	Da	Da	Da	Da	Nu	Nu	Nu
Suportă delegarea	Da	Nu	Nu	Da	Se poate face	Da	Da
Necesită sisteme client şi server Win2000?	Nu	Da	Nu	Da	Nu	Nu	Nu
Datele utilizatorilor trimise în text clar (necesită SSL)?	Da	Nu	Nu	Nu	Nu	Da	Nu
Suport pentru alte browser-e în afară de Internet Explorer?	Da	Nu	Nu	Nu	Da	Da	Da

Concluzii

Alegerea mecanismelor de autentificare şi autorizare potrivite pentru o anumită aplicație nu este un lucru uşor. Pentru aceasta trebuie luate în considerare mai multe procese între care se numără identificarea resurselor, alegerea unei strategii de autorizare, alegerea identităților pentru accesul la resurse, considerarea fluxului identităților, alegerea unei modalități de autentificare şi alegerea modalității de transmitere a identității utilizatorilor. În ceea ce priveşte mecanismele de autentificare, trebuie luate în considerare identitățile, depozitarea şi transmiterea acestora cât şi tipul browser-ului utilizat. Cel mai adecvat mecanism de autorizare pentru aplicațiile web este cel bazat pe roluri, datorită flexibilității acestuia, atât în ceea ce priveşte sursele din care se pot extrage rolurile sau apartenența la grupuri, cât şi datorită multitudinii de opțiuni potrivite celor mai diverse scenarii.

Bibliografie

1. Connell, J. – *Coding Techniques for Visual Basic .NET*, Microsoft Press, 2002
2. Jorgensen, D. – *Developing .NET Web Services with XML*, Syngress Publishing Inc., 2002
3. Thai, T.; Lam, H. - *.NET Framework Essentials, 2nd Edition*, O'Reilly, 2002
4. *** - *Proiectare în Windows.NET* – NetReport nr. 5/2001
5. Securitatea datelor – PcMagazine România, aprilie 2002