# Securitatea aplicatiilor ASP.NET

Catalin MAICAN
Universitatea *Transilvania* din Brasov

*Web applications have to ensure a security that would permit their users to do their job or transaction safely. ASP.NET web applications do not make a separate part of these and, by using new authorization and authentication techniques brought by .NET Framework, the implementation of good security can be present in every application and in every programmer's hands.*

***Keywords:*** *security, tiers, security architecture, authentication, authorization.*

## Introducere

Arhitectura logica a unei aplicatii necesita ca orice sistem sa fie compus din servicii care coopereaza, grupate în urmatoarele niveluri: servicii utilizatori, servicii de afaceri si servicii de date. Valoarea arhitecturii este data de identificarea tipurilor generice de servicii prezente de obicei în orice sistem, pentru a asigura segmentarea corecta si pentru a conduce definitia interfetelor între parti. Segmentarea permite alegeri de arhitectura si de design mai discrete în momentul implementarii fiecarui nivel, în vederea construirii unei aplicatii usor de întretinut.

Nivelurile pot fi descrise astfel:

- *serviciile utilizator* sunt responsabile pentru interactiunea clientilor cu sistemul, pentru a pune la dispozitie o legatura comuna catre logica afacerii încapsulata de componentele din nivelul *Servicii de afaceri.* În mod traditional, serviciile utilizator sunt asociate cu utilizatorii interactivi. Totusi, acestea executa procesarea initiala de cereri catre alte sisteme în care nu exista interfete cu utilizatorii. În acest nivel sunt executate serviciile de autentificare si autorizare, servicii care depind de tipul clientului;
- *serviciile de afaceri* ofera functionalitatea de baza a sistemului si încapsuleaza logica afacerii, fiind independente de canalul de distributie si de sistemele back-end sau de sursele de date. Acest lucru ofera stabilitatea si flexibilitatea necesara evolutiei sistemului în vederea suportarii de noi canale sau de sisteme back-end. În mod normal, raspunsul la o cerere din acest nivel necesita cooperarea mai multor componente;
- *serviciile de date* ofera accesul la date (gazduite în interiorul granitelor sistemului) si catre alte sisteme prin interfete generice, usor de utilizat din nivel de *Servicii de afaceri*. Acest nivel abstactizeaza multitudinea de sisteme back-end si de surse de date, incluzând regulile specifice de acces si formatele de date.

Clasificarea logica a tipurilor de servicii poate sa fie corelata cu posibila distributie fizica a componentelor care implementeaza sistemul, dar este si relativ independenta de aceasta. De asemenea, nivelurile logice pot fi identificate la orice nivel de agregare, acest lucru însemnând faptul ca nivelurile pot fi identificate pentru un sistem ca întreg, în contextul interactiunilor cu mediul, cât si pentru subsistemele componente. De exemplu, fiecare nod la distanta care gazduieste servicii Web consta în *Servicii utilizator* (gestionarea cererilor si a mesajelor), *Servicii de afaceri* si *servicii de date*. Nivelurile logice descrise mai sus nu implica un numar specific de parti fizice. De exemplu, toate serviciile logice pot sa fie localizate din punct de vedere fizic pe acelasi calculator sau pot sa fie distribuite în mai multe calculatoare.

Un model des întâlnit pentru aplicatiile Web este acela de a localiza componentele de afaceri si componentele de acces la date pe serverul Web, acest lucru micsorând traficul în retea, crescând astfel performanta. În figura 1 serverul web se comporta si ca server de aplicatii.

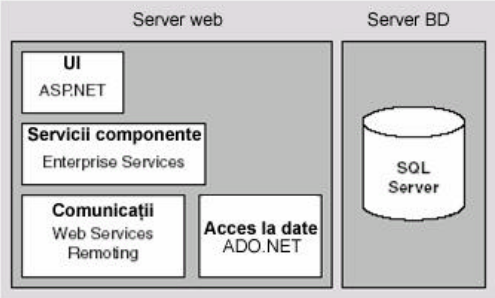


**Fig. 1.** Serverul Web în calitate de server de aplicatii

Un alt model este acela în care aplicatia de afaceri este separata, fiind utilizat mai ales în aplicatiile Internet în care serverul Web face parte dintr-un perimetru de retea (zona demilitarizata – DMZ), fiind separat de utilizatori si de serverele de aplicatii prin aplicatii de filtrare a pachetelor (Fig. 2).

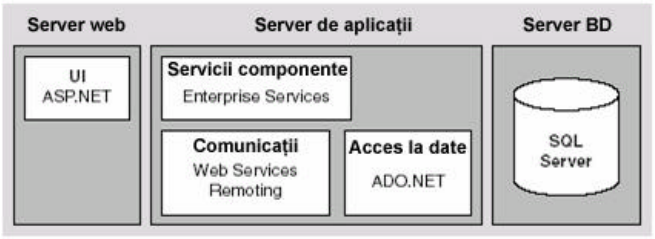


**Fig. 2.** Serverul de aplicatii separat

Aplicatiile Web .NET implementeaza unul sau mai multe servicii logice prin utilizarea urmatoarelor tehnologii:

- ASP.NET este utilizat pentru a implementa Serviciile utilizatorilor. Ofera o arhitectura extensibila care poate fi utilizata pentru a construi pagini de Web;
- Enterprise Services ofera servicii pentru aplicatii la nivel de infrastructura, printre care se numara tranzactiile distribuite, serviciile de management ale resurselor precum *object pooling* pentru componentele .NET;
- Serviciile Web permit schimbul de date si invocarea de la distanta a aplicatiilor utilizând schimburile de mesaje pe baza de SOAP pentru a transmite datele prin firewall-uri si sisteme eterogene;
- .NET Remoting ofera un cadru de lucru pentru accesarea obiectelor distribuite peste granitele proceselor si masinilor;
- ADO.NET si SQL Server 2000 ofera servicii de acces la date, fiind utilizat pentru accesarea obiectelor distribuite între ofertanti de date ADO.NET. A fost creat înca de la început pentru aplicatii Web distribuite, având si suport pentru scenariile deconectate, asociate cu aplicatiile Web. SQL Server ofera securitate integrata care utilizeaza mecanismele de autentificare ale sistemului de operare (Kerberos si NTLM). Autorizarea este oferita de logon si de permisiile granulare aplicate la nivel de obiecte individuale în baza de date;
- Internet Protocol Security (IPSec) ofera criptare punct-la-punct la nivel de protocol de transport, precum si servicii de autentificare;
- Secure Sockets Layer ofera canale de comunicare securizate punct-la-punct, în care datele trimise prin aceste canale sunt criptate.

**Arhitectura de securitate**

În

se poate observa modelul serverului de aplicatie separat împreuna cu serviciile de securitate oferite de diverse tehnologii introduse mai sus. Autentificarea si autorizarea are loc în mai multe puncte individuale, în fiecare nivel (tier). Serviciile sunt oferite în principal de Internet Information Services (IIS, ASP.NET, Enterprise Services si SQL Server). Canalele securizate de comunicatie, utilizând o combinatie între SSL si IPSec, sunt aplicate prin intermediul nivelurilor, pornind de la clienti (browser sau alte dispozitive) pâna la baza de date.

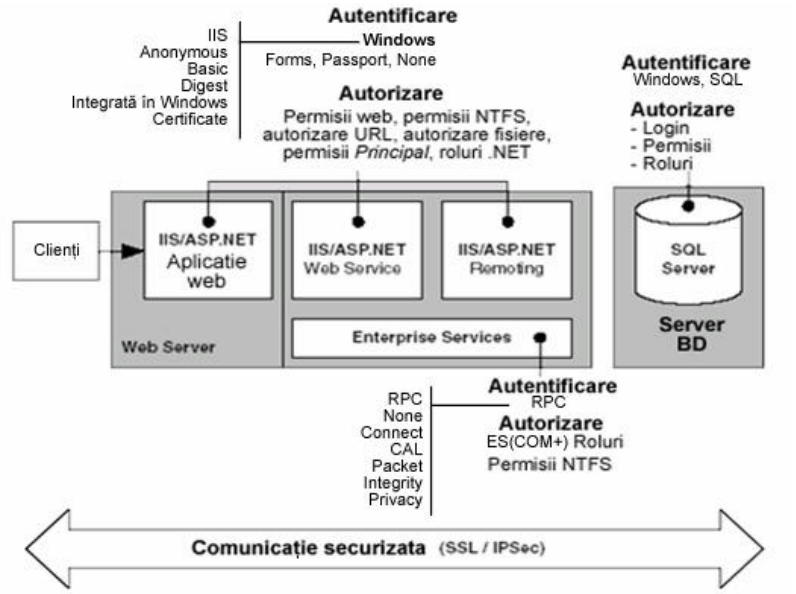


**Fig. 3.** Arhitectura de securitate

Facilitatile de autentificare, autorizare si comunicatie securizata pentru fiecare tehnologie în parte, pot fi observate în Tabelul 1.

Tabelul 1 - Facilitati de comunicare, autentificare si autorizare

| Tehnologia | Autentificare | Autorizare | Comunicatie securizata |
|---|---|---|---|
| IIS | - Anonynous<br>- Basic<br>- Digest<br>- Integrata în Windows (Kerberos sau NTLM)<br>- Certificate | - Restrictii la nivel de adrese IP/DNS;<br>- Permisii NTFS;<br>- Liste de control al accesului (ACL) în Windows pentru fisierele cerute; | - SSL; |
| ASP.NET | - Nici una (Custom);<br>- Windows;<br>- Forms;<br>- Passport. | - Autorizarea la nivel de fisier;<br>- Autorizarea la nivel de URL;<br>- Permisii de tip *Principal*;<br>- Roluri .NET. | |
| Servicii Web | - Windows;<br>- Nici una (Custom);<br>- Autentificare la nivel de mesaj. | - Autorizarea la nivel de fisier;<br>- Autorizarea la nivel de URL;<br>- Permisii de tip *Principal*;<br>- Roluri .NET. | SSL si criptare la nivel de mesaj. |
| .NET Remoting | - Windows. | - Autorizarea la nivel de fisier;<br>- Autorizarea la nivel de URL;<br>- Permisii de tip *Principal*;<br>- Roluri .NET. | SSL si criptare la nivel de mesaj. |
| Enterprise Services | - Windows. | - Roluri Enterprise Services (COM+);<br>- permisii NTFS. | - criptare la nivel de *Remote Procedure Call* (RPC). |
| SQL Server 2000 | - Windows (Kerberos / NTLM);<br>- autentificare SQL Server. | - login la nivel de server;<br>- login la nivel de baza de date;<br>- roluri fixe în baza de date;<br>- roluri definite de utilizatori;<br>- roluri la nivel de aplicatie;<br>- permisii la nivel de obiect. | - SSL. |
| Windows 2000/ .NET | - Kerberos;<br>- NTLM. | - Liste de control al accesului (ACL) din Windows. | IPSec. |

**Autentificarea**

.NET Framework are urmatoarele optiuni de autentificare: modurile de autentificare ASP.NET; autentificarea de tip Enterprise Services; autentificarea de tip SQL Server. Modurile de autentificare ASP.NET cuprind autentificarea de tip Windows, Forms, Passport si None (nici un fel de autentificare).

- *Autentificarea de tip Windows*. Prin acest tip de autentificare, ASP.NET se bazeaza pe IIS pentru a autentifica utilizatorii si pentru a crea jetonul de acces care sa reprezinte identitatea autentificata. IIS ofera urmatoarele mecanisme de autentificare:

- *Basic* – necesita ca utilizatorii sa ofere datele de identificare sub forma unui nume de utilizator si a unei parole, pentru a-si putea dovedi identitatea, mecanism de autentificare bazat pe standardul Internet propus de RFC 2617. Atât Netscape Navigator cât si Internet Explorer suporta acest tip de autentificare. Datele introduse de utilizatori sunt transmise de la browser-ul Web catre server într-un format necriptat, bazat pe formatul de codare Base64. Deoarece serverul Web obtine datele de identificare în mod necriptat, acesta poate sa execute apeluri la distanta, utilizând date introduse de utilizator. Aceasta modalitate de autentificare ar trebui utilizata numai împreuna cu un canal de comunicatie securizat (SSL), aplicata tuturor paginilor, deoarece datele de identificare sunt transmise la fiecare cerere;
- *Digest* – autentificare introdusa de IIS 5.0, este similara cu autentificarea de tip *Basic*, cu exceptia faptului ca datele transmise de utilizator sunt transmise sub forma unei valori *hash*. Desi este mult mai sigura decât autentificarea de baza, necesita Internet Explorer 5+ si o configurare specifica a serverului;
- *Autentificare integrata în Windows* – (Kerberos sau NTLM, în functie de client si de configuratia serverului) utilizeaza un schimb de date criptat cu browser-ul Web al utilizatorului pentru a confirma identitatea acestuia. Este un tip de autentificare suportat numai de Internet Explorer, nu si de Netscape Navigator, în consecinta tinzând sa fie utilizata numai la nivel de intraneturi, în care clientii software pot fi controlati. De asemenea, este utilizata de serverul Web în cazul în care accesul anonim este dezactivat sau daca accesul anonim este interzis prin permisiile la nivel de Windows;
- *Certificate* – utilizeaza certificate la nivel de clienti pentru a identifica în mod pozitiv un client. Certificatul clientului este transmis de catre browser (sau alta aplicatie) catre serverul Web. În cazul serviciilor Web, serviciul Web client transmite certificatele prin utilizarea proprietatii *ClientCertificates* din obiectul *HttpWebRequest*. Apoi, serverul Web extrage identitatea utilizatorului din certificat, putând-o mapa la un cont Windows. Acest tip de autentificare necesita ca certificatele sa fie instalate pe calculatoarele clientilor, de aceea fiind utilizata mai mult în scenarii de tip intranet sau extranet, în care populatia clientilor este bine-cunoscuta si controlata;
- *Anonima* – în cazul în care nu se doreste autentificarea clientilor (sau se implementeaza scheme proprii de autentificare), IIS poate fi configurat pentru a accepta toti clientii. În acest caz, serverul Web creeaza un jeton de acces Windows care reprezinta toti utilizatorii anonimi. Contul implicit pentru acces anonim este IUSR_NumeMasina, în care NumeMasina este numele NetBIOS al calculatorului, specificat la instalare.

- **Autentificarea de tip Passport**. Prin aceasta modalitate de autentificare, ASP.NET utilizeaza serviciul centralizat de autentificate Microsoft Passport. ASP.NET ofera o modalitate simpla de acces la aceasta functionalitate, oferita de Microsoft Passport Software Development Kit (SDK), pachet care trebuie instalat pe serverul Web.
- **Autentificare de tip Forms** utilizeaza redirectarea la nivel de client pentru a trimite utilizatorii neautentificati catre un formular HTML care le permite trimiterea catre server a datelor de identificare (nume de utilizator si parola). Datele de identificare sunt apoi validate, fiind generat în schimb un tichet care este returnat clientului. Tichetul de autentificare mentine identitatea utilizatorului precum si, în mod optional, o lista de roluri în care utilizatorul este membru, pe durata unei sesiuni. Acest tip de autentificare poate fi utiliza-

ta pentru personalizarea site-urilor Web, în acest caz fiind necesar putin cod, deoarece ASP.NET gestioneaza procesul în mod automat, prin simple configuratii. De asemenea, cookie-ul utilizat trebuie sa contina numai numele utilizatorului. Acest tip de autentificare trimite numele de utilizator si parola catre serverul Web sub forma de text în clar, pentru securitate trebuind utilizat si un canal securizat de SSL. Pentru protectia cookie-ului de autentificare transmis la fiecare cerere, ar trebui utilizat SSL pentru toate paginile unei aplicatii, nu numai pentru pagina de login.

- **Autentificare a de tip None** indica fie faptul ca nu se doreste autentificarea utilizatorilor, fie utilizarea unei modalitati proprii de autentificare.
- **Autentificare de tip Enterprise Services** este executata prin utilizarea infrastructurii de transport a Remote Procedure Call (RPC), care la rândul ei utilizeaza *Security Service Provide Interface* a sistemului de operare. Clientii aplicatiilor Enterprise Services pot fi autentificati prin Kerberos sau prin NTLM. O componenta servita poate fi gazduita într-o biblioteca sau într-un server de aplicatie. Bibliotecile de aplicatie sunt gazduite în procesele clientilor, asumându-si identitatea clientilor. Aplicatiile server ruleaza în procese separate, sub propria lor identitate. Apelurile catre o componenta servita pot fi autentificare la urmatoarele niveluri:

- *Default* – este utilizat nivelul implicit de autentificare pentru pachetul de securitate;
- *None* – nu are loc nici o autentificare;
- *Connect* – autentificarea are loc numai la efectuarea conexiunii;
- *Call* – autentificarea are loc la fiecare apel de procedura la distanta;
- *Packet* – autentifica si verifica daca toate datele de apel sunt receptionate;
- *Packet Integrity* – autentifica si verifica daca datele au fost modificate în tranzit;
- *Packet Privacy* – autentifica si cripteaza un pachet, inclusiv datele si identitatea si semnatura expeditorului.

- **Autentificarea SQL Server** poate autentifica utilizatorii prin folosirea autentificarii de tip Windows (NTLM sau Kerberos) sau poate sa-si construiasca propria schema de autentificare, cunoscuta sub numele de autentificare SQL. Sunt disponibile doua optiuni:

- SQL Server si Windows – clientii se pot conecta la o instanta SQL Server fie prin utilizarea autentificarii SQL Server, fie prin utilizarea autentificarii de tip Windows. Aceasta optiune este cunoscuta si sub numele de *mod mixt de autentificare*;
- Numai Windows – utilizatorii trebuie sa se conecteze la instantele SQL Server utilizând numai autentificarea de tip Windows.

**Autorizarea**

.NET Framework ofera urmatoarele optiuni de autorizare: optiunile ASP.NET; autorizarea Enterprise Services; autorizarea SQL Server. Optiunile de autorizare ASP.NET pot fi utilizate de aplicatiile web ASP.NET, serviciile Web si de componentele la distanta. ASP.NET ofera urmatoarele optiuni de autorizare:

- *URL Authorization* este un mecanism de autorizare, configurat de setarile la nivel de masina si la nivel de fisiere de configuratie ale aplicatiei. Autorizarea permite restrictionarea accesului la fisiere si directoare în spatiul de nume *Uniform Resource Identifier* (URI) într-o aplicatie. De exemplu, se poate interzice sau permite în mod selectiv accesul utilizatorilor nominalizati la anumite fisiere si foldere (adresate pe baza unui URL). De asemenea, se poate restrictiona accesul pe baza rolului sau pe baza comenzii HTTP utilizata (GET, POST etc.). Acest tip de autorizare necesita o identitate autentificata;
- *File Authorization* se aplica numai în cazul în care se utilizeaza mecanismele de autentificare oferite de IIS. Se poate utiliza pentru a restrictiona accesul la anumite fisiere de pe serverul Web. Permisiile de acces sunt determinate de ACL atasate fisierelor;
- *Principal Permission Demands* poate fi utilizat sub forma unui mecanism aditional de control fin al accesului. Se permite astfel controlul accesului la clase, metode sau chiar blocuri de cod individuale, pe baza identitatii si apartenentei la un grup pentru utilizatorii individuali;

- *Rolurile .NET* sunt utilizate pentru a grupa utilizatorii care au aceleasi permisii într-o aplicatie, fiind asemanatoare din punct de vedere conceptual cu implementarile anterioare, de exemplu grupurile de utilizatori din Windows sau rolurile COM+. Spre deosebire acestea, rolurile .NET nu necesita identitati Windows autentificate, putând fi utilizate si cu scheme de autentificare pe baza de tichete, precum autentificarea de tip *Forms*. Pot fi utilizate pentru a controla accesul la resurse si operatii, putând fi configurate atât în mod declarativ cât si în mod programatic.

Autorizarea Enterprise Services – accesul la functionalitatea continuta în componentele servite în cadrul aplicatiilor Enterprise Services este guvernat de apartenenta la roluri. Aceste roluri sunt diferite de rolurile .NET si pot contine grupuri de utilizatori Windows sau conturi. Apartenenta la roluri este definita în interiorul catalogului COM+.

Autorizarea SQL Server permite utilizarea de permisii care pot fi aplicate la nivel de obiecte individuale din baza de date. Permisiile pot fi bazate pe apartenenta la roluri (SQL Server ofera roluri fixe în baza de date, roluri definite de utilizatori si roluri de aplicatie) sau pot fi acordate utilizatorilor individuali Windows sau grupurilor.

## Concluzii

Serviciile de securitate pot cuprinde mai multi pasi prin care unui utilizator i se poate permite accesul la o anumita resursa. Dintre acestia, cei mai importanti sunt *autentificarea*, prin care se dovedeste identitatea unui anumit utilizator si se face o legatura cu serviciile de securitate ale aplicatiei si *autorizarea*, prin care se poate permite utilizatorului autentificat accesul la resursa dorita. Autentificarea se poate face în mai multe locuri, în functie de nivelul de securitate dorit. De asemenea, autorizarea se poate face în toate nivelurile de acces, tot în functie de nivelul de securitate dorit. Dupa cum s-a putut observa, aplicatiile ASP.NET permit o gama larga de servicii de autentificare si autorizare, dar pentru dezvoltarea de aplicatii mari nu trebuie uitata nici arhitectura logica a aplicatiei, despartirea sau respectiv centralizarea nivelurilor (tiers) având atât beneficii cât si dezavantaje.

## Bibliografie

1. Bagnall, B.; Chen, P.; Goldberg, S. – *C# for Java Programmers*, Syngress Publishing Inc., 2002;
2. Connell, J. – *Coding Techniques for Visual Basic .NET*, Microsoft Press, 2002;
3. Grimes, F. – *Microsoft .NET from Programmers*, Manning Publications Co., 2002;
4. Ryan, D.; Ryan T. – *ASP.NET: Your visual blueprint for creating Web applications on the .NET Framework*, Hungry Minds, Inc. 2002;
5. Smith, P. – *Client/Server Computing Second Edition*, Sams Publishing, 1994;
6. Solomon, D.; Russinovich, M. - *Inside Microsoft Windows 2000, Third Edition* - Microsoft Press, 2000.