Securitatea agentilor mobili

Asist. Ana-Ramona LUPU
Catedra de Informatic a Economica, A.S.E. Bucuresti

Mobile agents develop as a new promising paradigm of distributed computation. Moving through the network, they are not secure, some security mechanisms need to be implemented in order to protect servers against the hostile actions of agents they are executing, as well as mobile agents against the tampering actions of the hosts. The second problem is much more difficult to solve. Distance servers that host mobile agents can initiate hostile attacks or analyses to determine agent's logic or to discover its accumulated data. Beside the existing hardware solutions, researchers try to find software solutions to protect mobile agents against the hosts that are executing their code.
Keywords*: mobile agent, security, non-detachable signature.*

Introducere

Agentii mobili sunt entitati software autonome care sunt capabile sa migreze în medii de executie diferite. Din cauza mobilitatii si autonomiei lor, conexiunile permanente nu mai sunt necesare, ceea ce face ca agentii mobili sa fie foarte potriviti pentru conexiuni cu latime de banda redusa si comunicatii asincrone. Si, lucru deloc neglijabil, ei ofera o alternativa pentru lucrul în medii eterogene. Datorita caracteristicilor lor, agentii mobili sunt ideali pentru aplicatiile de comert electronic în retele deschise. Un agent poate cauta anumite produse sau servicii si poate negocia din partea proprietarului sau cu alte entitati. Agentii mobili pot fi folositi si ca agenti de vânzari, dar ei sunt vulnerabili la atacuri, mai ales din partea gazdelor rau intentionate. În executarea codului mobil în medii lipsite de încredere apar trei mari probleme:

- daca agentul mobil poate sa se protejeze de alterari din partea unei gazde rauvoitoare (integritatea codului si a executiei);
- daca agentul mobil poate sa ascunda programul pe care vrea sa îl execute (confidentialitatea codului);
- daca agentul mobil poate sa semneze un document la distanta fara sa dezvaluie cheia privata a utilizatorului.

Pâna nu demult a existat convingerea ca vulnerabilitatea agentilor mobili putea fi împiedicata doar prin solutii hardware. Acesta convingere s-a dovedit a fi gresita, si este cunoscuta în literatura de specialitate ca *paradoxul lui Chess*, pentru ca Chess argumenta în 1995[1]: "*Este imposibil de prevenit atacul agentilor daca nu este disponibil un hardware de încredere (si rezistent la atacuri)…. Fara un astfel de hardware, cineva rauvoitor poate oricând sa atace agentul.*"

Adevarata autoaparare a agentilor este posibila doar daca exista un nucleu de operatii software pentru care integritatea si confidentialitatea calculelor pot fi demonstrate matematic. *Jakobson* si *Juels* [3] au propus ideea codului executabil care asociaza un utilizator unei tranzactii de plata. Utilizatorul asociaza codul cu un asa-numit certificat de negociere care garanteaza o cumparare limitata. Acest protocol este practic, dar nu foarte flexibil, pentru ca necesita un numar mare de certificate pentru anumite operatii de cumparare.

Pe de alta parte, solutiile criptografice pentru asigurarea securitatii executiei agentilor mobili trebuie sa tina seama de cerinta de a nu introduce protocoale interactive (care sa implice proprietarul agentului).

Probleme de securitate ale agentilor mobili

Paradigma agentilor software vine sa extinda posibilitatile oferite de modul traditional de comunicare la distanta si calcul distribuit, dar în acelasi timp ridica noi probleme de securitate. Acestea se împart în doua mari categorii: protejarea gazdelor de agenti rau intentionati si protejarea agentului de site-urile gazda.

Tabelul 1. Caracteristici de securitate pentru câteva sisteme bazate pe agenti mobili

Sistem	Comunicatii sigure	Protectia resurselor serverului	Protectia agentilor
Teles cript	Transferul agentilor este autentificat (folosind RSA) si este criptat (folosind RC4)	Acces la resurse pe baza capacitatii. Pot fi impuse norme. Autorizare bazata pe autoritatea agentului.	Nu suporta.
Tacoma	Nu suporta.	Nu suporta.	Nu suporta.
Agent Tcl	Foloseste PGP pentru autorizare si criptare	Foloseste ca mediu de executie sigur Safe Tcl. Nu ofera suport pentru autorizari bazate pe proprietar.	Nu suporta.
Aglets	Nu suporta.	Drepturi de acces precizate static, pe baza a doua categorii de securitate: de încredere si lipsit de încredere.	Nu suporta.
Voyager	Nu suporta.	Programatorii trebuie sa extinda Security Manager. Doar doua categorii de securitate: nativ si strain.	Nu suporta.
Concordia	Transferul agentilor este criptat si autentificat folosind protocolul SSL.	Security Manager minitorizeaza accesul folosind un ACL configurat statistic, pe baza identitatii proprietarului agentului.	Agentii sunt protejati de alti agenti prin mecanismul de acces la resurse.
Ajanta	Transferul agentilor este criptat si autentificat folosind protocoalele ElGamal si DSA.	Acces la resurse bazat pe capacitate. Autorizare pe baza proprietarului agentului.	Mecanisme de detectare a alterarii starii sau codului agentului.

Protejarea gazdelor de atacurile agentilor rau intentionati este posibila prin realizarea unui control eficient al accesului si mecanisme tip sandbox. O problema mai dificila este însa protejarea agentului de abuzuri din partea unui server ostil. Pe parcursul executiei sale, agentul se afla într-o relatie foarte asimetrica cu serverul, deoarece serverul trebuie sa acceseze codul datele si starea agentului pentru a-l putea executa. Nu este prea clar cum poate agentul folosi informatie privata fara ca aceasta sa fie dezvaluita mediului sau de executie. Este o problema foarte dificil de rezolvat. În lucrarea lui *Yi s.a.* [10] întâlnim chiar opinia conform careia: "*Consensul actual este ca este imposibil din punct de vedere computational ca agentii sa fie protejati de gazde rau intentionate. În loc sa se abordeze problema din punct de vedere computational (dificil), cercetarea curenta cauta mijloace moderne de a încuraja comportamentul corect al gazdelor*".

În tabelul 1 prezinta caracteristicile de securitate pentru câteva dintre cele mai cunoscute sisteme bazate pe agenti mobili. Exista mai multe tipuri de atacuri din partea gazdei:

- poate sa distruga agentul, afectând astfel functionarea aplicatiei sale parinte;
- poate sa fure informatii utile depozitate de agent, rezultate intermediare obtinute în deplasarile sale;
- poate sa modifice datele transportate de agent;
- poate încerca sa altereze codul agentului si sa îl determine sa realizeze actiuni negative când se reîntoarce pe site-ul parinte. Acest lucru este deosebit de periculos, deoarece site-ul parinte îsi trateaza proprii agenti ca entitati de încredere si probabil ca le permite sa sara peste mecanismele de control ale accesului la resurse.

Eforturile cercetarilor pentru rezolvarea acestei probleme merg în doua directii:

A. Detectarea alterarii agentilor. Aceasta directie are ca scop detectarea alterarii agentilor *a posteriori*, urmarirea identitatii gazdei ilegitime si dovedirea comportamentului sau. Odata ce a fost detectat un atac din partea unui server, aceasta informatie poate fi trimisa unei agentii de rating care sa mentina înregistrari despre nivelul de încredere al serverelor (control social). *Vigna s.a. [9]* au introdus un mecanism de urmarire care înregistreaza executia unui agent si interactiunea sa cu mediul de executie. Mecanismul de urmarire conduce la descoperirea gazde vi-

novate. *Yi[10]* propune un Centru de serviciu al agentilor care sa urmareasca itinerarul unui agent. *Kotzanikolaou s.a*. [4] folosesc un sistem multiagent care poate detecta agentii mobili care au fost victimele atacurilor unor gazde. O solutie propusa de *Meadows[6]* este de a adauga date unui agent care sa fie oferite ca posibile tinte ale alterarii, iar la întoarcerea agentului sa se verifice daca aceste date au fost sau nu modificate.

Desi aceste solutii se pot dovedi suficiente în unele cazuri, detectarea *a posteriori* nu este eficienta pentru atacuri în care vinovatul nu mai poate fi identificat sau nici nu mai exista în momentul detectarii fraudei.

B. Prevenirea alterarii agentilor. În acest caz se merge pe ideea prevenirii alterarii agentilor *a priori*. Exista doua tipuri de prevenire: activa sau pasiva. În cazul mecanismelor de **prevenire pasiva** agentii sunt protejati prin solutii de tip organizational sau arhitectural. *Farmer s.a*. [2] propun o schema în care agentii pot sa circule doar în medii de executie sigure. Astfel, se realizeaza fixarea unei retele sigure de noduri, criptarea agentilor pentru transmiterea lor de la un nod la altul si autentificarea gazdei înainte ca agentul sa se deplaseze spre el, si de asemenea identificarea agentului de catre gazda. Este însa puternic afectat conceptul de sistem deschis, pentru care un nou server se poate adauga oricând apar noi nevoi suplimentare. *Merwe si Sholms[7]* introduc un sistem de comert bazat pe agenti, unde agentii sunt implementati cu obiecte distribuite care comunica la distanta. Aceste abordari fie se bazeaza foarte mult pe încrederea pe care o putem avea într-o gazda, fie compromit avantajele agentilor mobili, cum sunt autonomia sau migratia.

Prevenirea activa are în vedere dezvoltarea de solutii care ofera agentului protectie fata de atacurile unor gazde ostile, fara însa a compromite avantajele paradigmei agentilor mobili. Poate fi vorba de dispozitive hardware, dar utilizarea lor este redusa datorita costurilor ridicate. Pe de alta parte, în ultima vreme se poate vorbi de mecanisme de prevenire activa bazate pe software. O prima abordare în prevenirea activa bazata pe software pleaca de la ideea de a face atacul agentilor mobili cât mai dificil si mai scump. Se pot aplica tehnici speciale pentru a face codul ilizibil, sau algoritmi de codare, solutii care pot fi utile daca este demonstrata eficienta lor, altfel fiecare noua tehnica introdusa este urmata imediat de contramasuri. Securitatea acestei metode nu poate fi demonstrata. *Loureiro s.a*. *[5]* propun o metoda care permite trimiterea de cod mobil pe o gazda care nu e de încredere si evaluarea unei functii booleene criptate, mentinând confidentialitatea acestei functii. Dar, folosind aceasta schema nimeni în afara de proprietarul agentului nu va putea decripta rezultatele, pentru ca decriptarea implica cheia privata a acestuia.

Semnaturile nedetasabile au fost propuse de *Sander si Tschudin[8]*. Este folosita o tehnica denumita Computing with Encrypted Functions (CEF), în care un utilizator care foloseste un agent mobil trebuie sa utilizeze o functie semnatura s. Pentru a proteja functia s, utilizatorul o cripteaza cu o functie f, obtinând f_{semnat}:= sof si paseaza perechea de functii $(f(.), f_{semnat}(.))$ agentului ca parte a codului sau. Pe serverul pe care migreaza agentul se executa perechea $(f(.), f_{semnat}(.))$ pentru intrarea x si se obtine perechea $f(x)=m$ si $f_{semnat}(x)=s(f(x))=s(m)$. Asadar perechea de functii $(f(.), f_{semnat}(.))$ permite utilizatorului sa creeze mesaje semnate pe server, fara ca gazda sa aiba acces la s. Securitatea metodei provine din dificultatea descompunerii functiei criptate. Având în vedere ca ideea de baza a agentilor mobili este de a functiona cât mai autonom, autorii au explorat posibilitatile de aplicare neinteractiva a metodei lor. Chiar daca au fost gasite clase de functii candidate (au fost propuse functiile birationale), nu s-au gasit înca astfel de functii de criptare pentru care sa se poata demonstra securitatea.

Cercetarile în aceasta directie au fost continuate de *Kotzanikolaou s.a*. [4], care preiau schema CEF a semnaturilor nedetasabile dar folosesc functii exponentiale ca functii de criptare. Solutia pe care o ofera este bazata pe schema de criptare RSA, oferind o securitate egala cu a sistemului de criptare RSA.

În figura 1 este prezentata schematic solutia pentru cazul unui client care doreste sa faca cumparaturi de la un magazin electronic, unde: C - identificator client; d - cheia secreta a clientului; n, e - cheia publica a clientului (conform RSA); res_C - restrictiile clientului; bid_S - oferta serverului.

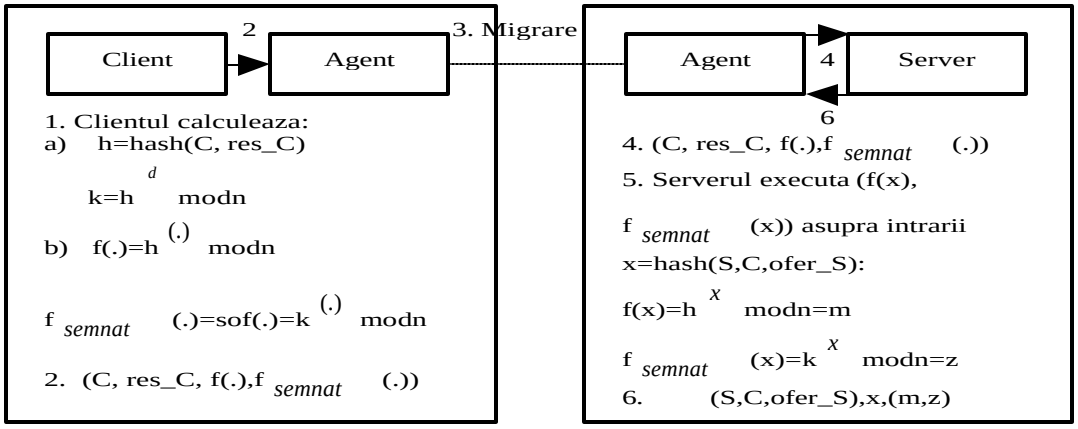

Fig. 1. Schema de semnatura digitala sigura bazata pe RSA si functii exponentiale

Desi ofera securitate, aceasta solutie nu ofera si confidentialitate, datele trimise prin retea putând fi citite de oricine asculta reteaua. Pentru asigurarea confidentialitatii, clientul ar trebui sa cripteze codul si datele agentului cu cheia publica a serverului, iar serverul la rândul lui ar trebui sa cripteze rezultatele executiei agentului cu cheia publica a clientului. Pe de alta parte, schema este asimetrica, si asociaza doar clientul tranzactiei, nu si serverul. Pentru realizarea simetriei ar fi nevoie ca si serverul sa autentifice tranzactia.

Necesitatea unor solutii rezistente la atacuri

Exista cazuri în care pentru utilizarea agentilor mobili este necesara o prevenire sigura demonstrata. De exemplu, detectarea *a posteriori* a atacurilor nu este potrivita în cazul banilor electronici, în cazul în care este vorba de sume prea mici pentru a justifica actiuni în justitie sau aplicarea legii este dificila. Semnarea digitala la distanta a contractelor ar deveni de asemenea imposibila deoarece divulgarea cheii private a utilizatorului îi rapeste acestuia posibilitatea de a dovedi ca nu el este cel care a comandat un anumit produs. Asadar, daca utilizatorul doreste sa delege agentului sau mobil inclusiv actiuni de acest tip, atunci prevenirea atacurilor si garantarea confidentialitatii sunt absolut obligatorii.

Dupa cum s-a mentionat, a existat pentru multa vreme în sânul comunitatii agentilor mobili convingerea ca o entitate care executa un anumit program are control deplin asupra executiei sale si poate întelege în întregime programul respectiv, motiv pentru care l-ar si putea modifica în orice mod doreste. Pentru verificarea acestei prezumtii nu exista argumente riguroase, dar exista câteva argumente intuitive:

- datele pot fi citite si schimbate în clar;
- pot fi manipulate programe în clar;
- mesajele în clar pot fi falsificate.

Pe de alta parte, nu exista nici un motiv teme inic pentru care programele trebuie sa fie executate în clar. Asadar ideea despre vulnerabilitatea agentilor mobili este probabil gresita, deoarece pleaca de la presupunerea ca un agent consta în date în clar si programe în clar.

Posibilitatea de executare a unor programe criptate ar conduce automat la confidentialitatea si integritatea codului în sensul ca anumite tipuri de atacuri nu mai sunt posibile. Atacurile din partea unor gazde rauvoitoare s-ar reduce la actiuni la suprafata agentului, cum ar fi: refuzul serviciilor, modificari aleatoare ale programului sau iesirilor sale.

Bibliografie

1. Chess, D., Grosof, B., Harrison, C., Levine, D., Parris, C si Tsudik, G.: *Itinerant Agents for Mobile Computing*, Technical Report (1995), IBM T. J. Watson Research Center, NY;

2. Farmer, W., Gutmann, J., Swarup, V.: *Security for Mobile Agents: Authentication and State Appraisal*, Proceedings of the European Symposium on Research in Computer Security (ESORICS), Springer-Verlag (1996), p 118-130;

3. Jakobson, M., Juels, A.:X-cash: *Executable Digital Cash*, Springer-Verlag (1998), p. 16-27;

4. Kotzanikolaou, P., Katsirelos, G., Chrissikopoulos, V.: *Mobile Agents for Secure Electronic Transactions. Recent Advances in Signal Procesing and Communications*, World Scientific and Engineering Society Press (1999), p 363-368;

5. Loureiro, S., Molva, R.: *Privacy for Mobile Code*, Proceedings of Distributed Object Security Workshop OOPSLA '99, Denver (1999);

6. Meadows, C., *Detecting Attacks on Mobile Agents*, Proceedings of DARPA Workshop on Foundations for Secure Mobile Code, Monterey, USA(1997);

7. Merwe, J., Sholms, S.H.,: *Electronic Commerce with Secure Intelligent Trade Agents*, Proceedings of ICICS '97, Denver (1999);

8. Sander, T., Tschudin, C.F.: *Protecting Mobile Agents Against Malicious Hosts. Mobile Agent Security*, Springer-Verlag (1998), p. 44-60;

9. Vigna, G.: *Cryptographic Traces for Mobile Agents. Mobile Agent Security*, Springer-Verlag (1998), p. 137-153;

10. Yi, X., Wang, X.F., Lam, K.Y.: *A Secure Intelligent Trade Agent System*, Proceedings of the International IFIP/GI Working Conference, TREC '98, Springer-Verlag (1998), p. 218-228.