

XMLRPC – o cale de a ieși din izolare. Un raport despre experiențele câștigate cu RSA SecurID si XMLRPC

Data: 07.01.2002

Autor: Florian Lindauer, SecureNet GmbH Munchen (fl@secure-net.de), tradus și adaptat de Liviu Marinescu, SecureNet S.R.L. (lm@secure-net.ro)

Compania: SecureNet S.R.L., Craiova, tel.: 051/410555, <http://www.secure-net.ro>

O componentă importantă a fiecărui serviciu Internet o reprezintă partea de autentificare a utilizatorilor (users). Într-un mediu în care cerințele de securitate sunt foarte ridicate, modelul usual “login + parolă” nu mai poate satisface aceste cerințe, chiar dacă transmisia datelor se face folosind SSL (Secure Socket Layer). Cu ajutorul protocolului SSL datele de autentificare a utilizatorului sunt transmise codificate, ele putând fi totuși spionate. O prima soluție ar fi parolele de unică folosință. Ele sunt valabile doar pentru o singură utilizare și după folosire nu mai sunt de nici un ajutor pentru potențialii agresori. Dacă parolele de unică folosință sunt emise câteva secunde înainte de utilizarea lor, atunci riscul de a fi spionate sau folosite împotriva voinței posesorului poate fi exclus. O soluție matură pentru rezolvarea problemelor de genul celor discutate mai sus o constituie produsul SecurID (vezi tabelul 1) al Companiei RSA (<http://www.rsasecurity.com/>). Firma SecureNet (<http://www.secure-net.de>) a folosit această tehnologie în proiectul MCdialog.net (<http://www.mcdialog.net>).

Tabelul 1. RSA SecurID

Soluția “SecurID” se bazează pe două lucruri distincte: un token hardware (de genul unei cărți de credit – să îl numim SecurID-card sau pe scurt card) și un secret (o parola și/sau un număr de pin). Cardul dispune de un display care afișează o parte integrantă din parolă în forma unui cod (alfa)numeric ce se schimbă din minut în minut. Utilizatorul folosește acest cod precum și pin-ul pe care numai el îl știe. Numai cine are atât pin-ul cit și codul parolei se poate autentifica: spionarea numărului de pin nu folosește celui care nu deține și cardul de generare a codului parolei. Astfel, SecurID generează neîncetat parole de unică folosință care sunt valide doar un timp limitat. Modul de generare a parolei nu poate fi descifrat pe baza parolelor generate anterior. Diferit de modul de lucru al RSA (generarea unei Public-Key) este faptul ca numai modulul hardware (cardul) și serverul dețin Secret-Key, deci această cheie este protejată în totalitate. Totodată și codul generat pentru parolă va avea o lungime mult mai mică în comparație cu lungimea unei chei de genul Public-Key. Atâta timp cât Server-ul este protejat, nu exista nici un risc pentru aflarea Secret-Key. Din aceste motive această tehnologie este clasificată în cercurile de specialitate ca fiind foarte sigură.

Pentru autentificarea utilizatorilor cu SecurID trebuie instalat un server special - RSA ACE/Server, care verifică permanent pin-ul transmis și codul parolei. În momentul implementării proiectului MCdialog.net nu exista un API Java pentru acest server iar API-ul C era disponibil numai sub Solaris, neaplicabil însă pe o platforma Linux. Între timp se poate ca lucrurile să se fi schimbat, dar poziția inițială este adesea regăsită și în celelalte cazuri: cine dorește să folosească serviciul este restricționat de platforma, limbajul de programare și/sau protocolul de comunicare.

În cazul nostru am folosit serviciul "Autentificarea SecurID" pentru cerințele unei aplicații Servlet Java. Dincolo de funcțiile specifice administrării utilizatorilor, va fi folosit numai serviciul cu semnătura "boolean is_valid" (codul parolei pentru a face disponibil codul pin). Acest serviciu trebuie să fie disponibil pentru adresare direct din aplicația Java. Aici intră în scenă XMLRPC.

XMLRPC

XMLRPC este o tehnologie de apel de metode la distanță RPC (Remote Procedure Call) pentru aplicații distribuite pe bază de XML (Extensible Markup Language). XML reprezintă o bază pentru alte standarde Web noi, cum ar fi XHTML (succesorul HTML), fiind de fapt un meta limbaj ce depinde de scopul aplicației. Pentru XMLRPC s-a creat un limbaj apropiat, ușor de întreținut. XMLRPC folosește protocolul HTTP pentru transmiterea cererilor, astfel încât serverul XMLRPC poate fi realizat cu programe CGI, putându-se adapta la serverele web și la clienții HTTP actuali. Pe site-ul <http://www.xmlrpc.com> se găsesc specificații, documentații și implementări pentru XMLRPC. Comparativ cu alternative ca Microsoft DCOM sau CORBA, XMLRPC este net mai simplu. Există asemănări cu SOAP (specificația pentru SOAP se găsește pe site-ul W3C la <http://www.w3.org/TR/SOAP>). SOAP este mai complex decât XMLRPC și probabil că îi va prelua funcționalitatea în viitor.

XMLRPC (<http://www.xmlrpc.com/>) reprezintă un protocol standard de comunicare pentru folosirea la distanță a diverselor servicii web. Pentru acest protocol există implementări pe toate platformele și pentru aproape toate limbajele de programare. Printre acestea se află limbaje larg răspândite ca Java sau C(++). Exemplul următor prezintă o secvență de cod de bază pentru realizarea autentificării serverului cu o interfață XMLRPC în C ca și descrierea folosirii sale în Java.

Serverul

Serverul SecurID ACE rulează pe o platformă Solaris care este conectată prin rețea cu un server web sub Linux. Pe Solaris este disponibil API-ul C ce poate furniza funcționalitatea dorită pentru serverul RSA. Pe Linux se creează un mic program server (daemon), care în caz contrar ar rula pe Solaris, și care poate fi adresat prin intermediul interfeței XMLRPC.

Exemplu:

```
static xmlrpc_value *
checkLogin (xmlrpc_env *env, xmlrpc_value *param_array, void *user_data)
{
xmlrpc_int32 status=0;
char *login,*pin,*passcode;
// get input parameters
xmlrpc_parse_value(env, param_array, "(ssss)", &login, &pin, &passcode);
if (env->fault_occurred) return NULL;
// validity check against RSA ACE Server
status=checkPasscode(login, passcode, pin);
// send result
return xmlrpc_build_value(env, "{s:i}", "status", status);
}
int main (int argc, char **argv)
{
// Prepare request handling
xmlrpc_server_abys_init(XMLRPC_SERVER_ABYSS_NO_FLAGS, "conf/abyss.conf");
// Add all the xmlrpc-methods we provide
```

```
xmlrpc_server_abysse_add_method_w_doc(
"securid.checkLogin", &checkLogin, NULL, "i:sss",
"Check validity of login, pin and passcode at the current time");
// Process requests
xmlrpc_server_abysse_run();
}
```

Clientul

Folosirea serviciului cu Java devine simplă. Este nevoie de unul din numeroșii clienți XMLRPC Java pentru apelarea funcțiilor disponibile. Mai jos este prezentat fragmentul de cod.

```
public int checkSecurId(String user, String passcode, String pin)
throws SomeSecurityException
{
try
{ XmlRpcClient server = new XmlRpcClient(XMLRPC_RSAACE_URL);
6 Helma XMLRPC-Implementierung für Java: xmlrpc.helma.org
Vector params = new Vector();
params.addElement(user);
params.addElement(pin);
params.addElement(passcode);
Hashtable xresult=(Hashtable)server.execute("securid.checkLogin", params);
result=((Integer)xresult.get("status")).intValue();
return result;
}
catch(Exception e)
{ log.debug("SecurID-xmlrpc: Exception: "+e.getMessage());
throw new SomeSecurityException("system-failure: authentication");
}
}
```

Pentru apelarea metodei, Clientul generează un obiect Request (codat XML) care cuprinde metoda și parametrii doriți și care este transportat de protocolul HTTP.

```
<?xml version="1.0"?>
<methodCall>
<methodName>securid.checkLogin</methodName>
<params><param><value><string>somelogin</string></value></param>
<param><value><string>1234</string></value></param>
<param><value><string>987654</string></value></param>
</params>
</methodCall>
```

Răspunsul Serverului conține parametrii rezultați – în cazul eșecului autentificării avem status 1=Login invalid

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<params><param><value><struct><member>
<name> status </name>
<value><i4> 1 </i4></value></member>
</struct></value></param></params>
</methodResponse>
```

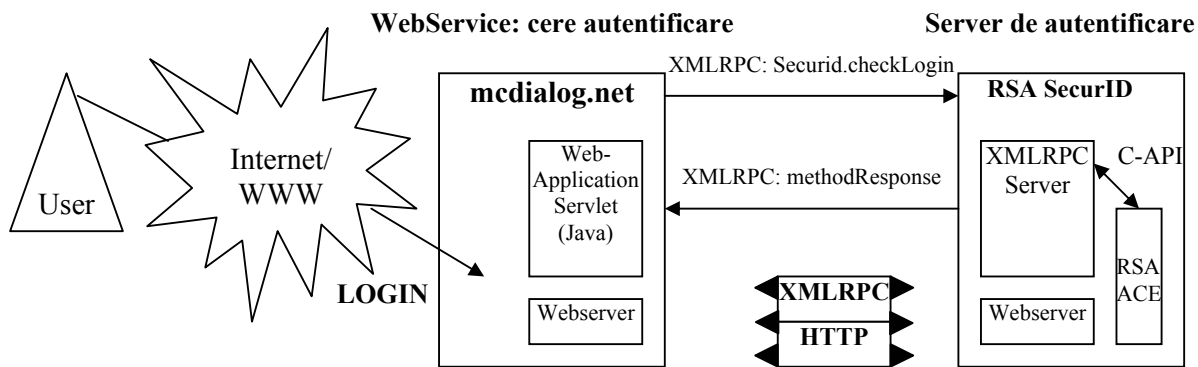


Fig. 1: Serviciu de autentificare disponibil via XML-RPC

Concluzii

XMLRPC este o tehnologie flexibilă, larg răspândită, puternică și în același timp simplu de integrat și folosit chiar în proiecte mari, în care părți ale acestora rulează distribuit în rețeaua proprie sau în internet. Din acest motiv această tehnică este de recomandat pentru împachetarea micilor servicii oferite de diverse firme în scopul de a le face disponibile acolo unde este nevoie de ele. Este de sperat, dar și de așteptat, ca în viitorul apropiat tot mai multe companii să își înzestreze produsele software cu interfețe către XMLRPC sau către tehnologii echivalente, de exemplu SOAP, tocmai în scopul unei intercomunicări necomplicate între diferitele componente software necesare în viața de zi cu zi a afacerilor moderne.

Referinte

- "Java and XML", Brett McLaughlin, O'Reilly, USA, June 2000
- MCDialog.Net – <http://www.mcdialog.net>
- RSA Security Inc. - <http://www.rsasecurity.com/>
- XML-RPC Home Page - <http://www.xmlrpc.org>
- XML-RPC for C and C++ - <http://xmlrpc-c.sourceforge.net>

Despre autor

Florian Lindauer lucrează de peste un an la SecureNet GmbH München ca și software developer pentru aplicații web în Java. Florian are experiență îndelungată în programare și administrare mai ales în sisteme Unix, dezvoltarea de programe fiind pentru el o pasiune.

