Nama : Alif Zidan Andriansyah

No.BP : 2111083004

Kelas : TRPL 3A

**RPC (Remote Procedure Calls)**

A. RPC (Remote Procedure Calls)

Remote Procedure Call (RPC) adalah protokol komunikasi yang memungkinkan program-program komputer untuk memanggil fungsi atau prosedur di ruang alamat memori yang berbeda, yang dapat berada di mesin yang berbeda dalam jaringan. RPC memungkinkan pemanggilan prosedur atau fungsi seperti yang dilakukan dalam pemrograman lokal, tetapi dengan memanfaatkan sumber daya yang terdistribusi secara geografis.

Protokol Message RPC memiliki dua (2) struktur yang berbeda, yaitu call message dan reply message. Tiap klien yang akan melakukan RPC pada suatu server di jaringan akan menerima balasan (reply) berupa hasil dari eksekusi prosedur tersebut. Dengan menggunakan spesifikasi yang unik untuk tiap prosedure remote, maka RPC dapat mencocokkan message balasan untuk tiap call message yang diminta klien. Dalam RPC, memiliki call message mengandung nilai-nilai unsigned integer yang digunakan untuk mengidentifikasi prosedur remote yang diminta. Nilai-nilai ini adalah :

1. Nomor Program
2. Nomor Versi dari Program
3. Nomor Prosedur

Reply message yang dikirimkan oleh server jaringan bervariasi tergantung apakah call messages yang diminta klien diterima atau ditolak. Reply message mengandung informasi yang digunakan untuk membedakan kondisi- kondisi yang diminta sesuai dengan call messages. Informasi ini antara lain :

1. RPM mengeksekusi call message dengan sukses
2. Implementasi remote tidak sesuai dengan protokol yang digunakan. Versi yang lebih rendah atau tinggi akan ditolak.
3. Program remote tidak tersedia pada sistem remote.
4. Program remote tidak mendukung versi yang diminta klien.
5. Nomor prosedur yang diminta tidak ada.

B. RPC memiliki fitur - fitur sebagai berikut: batching calls, broadcasting calls, callback procedures dan using the select subroutine.

a). Batching Calls

* mengijinkan klien untuk mengirim message calls ke server dalam jumlah besar secara sequence ( berurutan ). Batching menggunakan protokol streaming byte seperti TCP / IP sebagai mediumnya. Pada saat melakukan batching, klien tidak menunggu server untuk memberikan reply terhadap tiap messages yang dikirim, begitu pula dengan server yang tidak pernah mengirimkan messages reply. Fitur inilah yang banyak digunakan klien, karena arsitektur RPC didesain agar pada tiap call message yang dikirimkan oleh klien harus ada proses menunggu balasan dari server. Oleh karena itu maka pihak klien harus dapat mengatasi error yang kemungkinan terjadi karena pihak klien tidak akan menerima peringatan apabila terjadi error pada message yang dikirim.

b). Broadcasting Calls

* mengijinkan klien untuk mengirimkan paket data ke jaringan dan menunggu balasan dari network. FItur ini menggunakan protokol yang berbasiskan paket data seperti UDP/IP sebagai mediumnya. Broadcast RPC membutuhkan layanan port mapper RPC untuk mengimplementasikan fungsinya.

c). Callback Procedures

* mengijinkan server untuk bertindak sebagai klien dan melakukan RPC callback ke proses yang dijalankan oleh klien.

d). Menggunakan select Subrutin

* memeriksa deskripsi dari suatu file dan messages dalam antrian untuk melihat apakah mereka siap untuk dibaca (diterima) atau ditulis (dikirim), atau mereka dalam kondisi ditahan sementara. Prosedur ini mengijinkan server untuk menginterupsi suatu aktivitas, memeriksa datanya, dan kemudian melanjutkan proses aktivitas tersebut.

e). Otentifikasi RPC

* adalah proses yang digunakan untuk mengidentifikasi server dan klien pada RPC. Untuk setiap prosedur remote yang dilakukan protokol RPC menyediakan spot yang dipakai sebagai parameter otentifikasi yang berfungsi agar pemanggil (caller) dapat memberikan identitasnya kepada server. Parameter otentifikasi ini dibuat di paket klien. Otentifikasi RPC terdiri atas beberapa bagian.

C. Berikut ini adalah bagian- bagian pada otentifikasi RPC:

1. Protokol Otentifikasi RPC

Parameter yang digunakan adalah respon verifier. Sedangkan pada pihak klien, setiap paket RPC diberikan parameter otentifikasi dan parameter yang digunakan adalah credential danverifier.

1. Otentifikasi NULL

Digunakan pada sistem dimana pemanggil (caller) RPC tidak mengetahui identitasnya sendiri dan server tidak membutuhkan identitas pemanggil.

1. Otentifikasi UNIX

Jenis otentifikasi ini dibagi dua (2) yaitu otentifikasi pada sisi klien dan otentifiksi pada sisi server. Pada sisi klien, otentifikasi ini akan membuat otentifikasi handle dengan AIX permissions agar dapat berasosiasi dengan parameter credentials pada sistem UNIX. Sedangkan pada sisi server, server harus dapat menentukan tipe otentifikasi yang diberikan oleh pemanggil RPC.

1. Otentifikasi Data Encryption Standard (DES)

Membutuhkan keyserv daemon yang harus berjalan baik di sisi server maupun klien. Tiap pengguna pada sistem ini harus memiliki kunci publik (public key yang disahkan pada database kunci publik oleh Administrator jaringan tersebut.

1. Protokol Otentifikasi DES

Meliputi protokol penanganan DES pada proses otentifikasi RPC. Protokol ini mencakup 64-bit blok data DES yang terenkripsi dan menentukan panjang maksimum untuk user name pada jaringan yang digunakan.

1. Enkripsi Diffie-Hellman

Digunakan pada pembuatan kunci public pada otentifikasi DES dengan menggunakan 192-bit kunci. Enkripsi ini memiliki dua buah variable konstan, yaitu BASE dan MODULUS yang digunakan pada protokol otentifikasi DES.

Ada beberapa dari Bahasa yang digunakan RPC adalah meliputi:

1. Definition

File dengan bahasa RPC memiliki beberapa definisi, diantaranya adalah : enum,struct, union, typedef, const, dan program.

1. Structure

Struktur pada bahasa RPC dideklarasikan seperti pada pendeklarasian struktur dalam bahasa C.

1. Union

Union pada bahasa RPC berbeda dengan bahasa C. Kemiripan lebih ditunjukkan dengan variasi pada bahasa Pascal

1. Enumeration

Enumerasi pada bahasa ini memiliki syntax yang sama dengan bahasa C.

1. TypeDef

Tipe Definisi ( TypeDef ) pada bahasa ini memiliki syntax yang sama dengan typedef pada bahasa C.

1. Constant

Constant pada bahasa ini dapat digunakan jika variabel integer konstant dibutuhkan.

1. Programs

Program RPC dideklarasikan dengan syntax berikut secara berurutan : program definiton,version-list, version, procedure-list, procedure.

1. Declarations

Dalam bahasa ini, terdapat empat jenis tipe deklarasi yaitu : simple declarations, fixed-length array declarations, variable-length declaration, dan pointer declaration.

D. Prosedur Port Mapper

Program port mapper mendukung dua protokol, yaitu UDP dan TCP/IP. Program ini terhubung pada port 111 untuk kedua protokol ini. Berikut ini adalah prosedur-prosedur yang digunakan program port mapper pada kedua protokol ini :

1. NULL

Prosedur ini tidak berfungsi, prosedur ini tidak memberikan parameter dan juga tidak memberikan hasil.

1. SET

Prosedur ini akan meregister program pada port mapper dengan memberikan parameter sebagai berikut : program number (prog), version number (vers), transport protocol number (prot), dan nomor port yang diminta untuk layanan ini. Hasil dari prosedur ini berupa Boolean True atau False yang mengindikasikan sukses tidaknya proses mapping.

1. UNSET

Prosedur ini digunakan untuk me- unregister program pada port mapping jika program remote tidak lagi digunakan. Parameter yang dibawa sama dengan prosedur SET dikurangi nomor protokol dan nomor port.

1. GETPORT

Prosedur ini memberikan parameter berupa nomor program (prog), version number (vers), dan transport protocol number (prot) untuk mendapatkan hasil berupa nomor port untuk program yang diminta.

1. DUMP

Prosedur ini akan mencatat semua entri dalam database port mapper. Prosedur ini tidak membutuhkan parameter dan memberikan hasil berupa (prog), (prot), (vers), dan nomor port.

1. CALLIT

Prosedur ini digunakan untuk memanggil suatu program remote lain pada mesin yang sama tanpa harus mengetahui nomor port dari program yang diminta. Prosedur call adalah penggunaan stack untuk penyimpanan parameter dan alokasi ruang untuk variabel lokal. Prosedur call remote melibatkan pelaksanaan proses serupa, tetapi di sistem yang berbeda melalui jaringan. Dalam RPC, prosedur dipanggil dengan mengirimkan parameter ke server remote, menunggu balasan, dan mengeksekusi prosedur. Proses ini melibatkan stub di sisi klien dan server. Stub klien mengirim permintaan dan parameter, sementara stub server membaca parameter, menjalankan prosedur, dan mengirimkan hasil kembali.

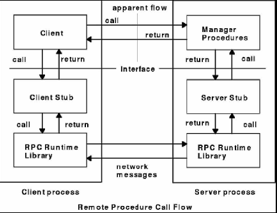


Diagram diatas memberikan gambaran mengenai flow dari eksekusi dalam proses RPC. Berikut ini adalah diagram yang akan menjelaskan secara rince mengenai proses yang terjadi pada klien dan server dalam eksekusi suatu prosedur RPC:

Berikut penjelasan dari diagram diatas :

1. Klien memanggil prosedur stub lokal. Prosedur Stub akan memberikan parameter dalam suatu paket yang akan dikirim ke jaringan. Proses ini disebut sebagai marshalling.
2. Fungsi Network pada O/S (Operating system - Sistem Operasi) akan dipanggil oleh stub untuk mengirim suatu message.
3. Kemudian Kernel ini akan mengirim message ke sistem remote. Kondisi ini dapat berupa connectionless atau connection-oriented.
4. Stub pada sisi server akan melakukan proses unmarshals pada paket yang dikirim pada network.
5. Stub pada server kemudian mengeksekusi prosedur panggilan lokal.
6. Jika eksekusi prosedur ini telah selesai, maka eksekusi diberikan kembali ke stub pada server.
7. Server akan melakukan proses marshals lagi dan mengirimkan message nilai balikan ( hasilnya ) kembali ke jaringan.
8. Message ini akan dikirim kembali ke klien.
9. Stub klien akan membaca message ini dengan menggunakan fungsi pada jaringan.
10. Proses unmarshalled kemudian dilakukan pada message ini dan nilai balikan akandiambil untuk kemudian diproses pada proses lokal.

Proses diatas akan dilakukan berulang- ulang ( rekursif ) dalam pengeksekusian RPC dalam suatu remote sistem.

E) Pencegahan Eksploitasi RPC

Pentingnya nilai informasi yang terkandung dalam suatu server atau sebuah komputer pribadi mengharuskan penggunanya baik user atau administrator harus melakukan tindakan preventif guna menghindari adanya kebocoran informasi ini. RPC sendiri adalah suatu layanan yang vital dan ada di dalam setiap sistem operasi, khususnya sistem operasi Microsoft Windows yang bergantung pada jenis layanan ini. Oleh karena itu menghilangkan layanan RPC pada sistem operasi terasa lebih susah ketimbang dilakukan pengawasan ekstra terhadap layanan ini. Berikut ini adalah beberapa cara yang dapat digunakan untuk melakukan hal ini ( dikutip dari buletin keamanan Microsoft MS03-026 ):

1. Memblokir port 135, 137, 138 dan 445 pada UDP dan port 135, 149, 445, dan 593 pada TCP melalui Firewall. Disfungsikan COM Internet Services ( CIS ) dan RPC melalui HTTP yang menggunakan port 80 dan 443 terutama pada jaringan remote yang menggunakan VPN ( Virtual Private Network ) atau sejenisnya.
2. Gunakan personal Firewall seperti Internet Connection Firewall.
3. Blokir semua port sering diekploitasi dengan menggunakan filter IPSEC.
4. Disfungsikan fitur DCOM pada setiap komputer atau server. Disable DCOM on all affected machines
5. Khusus Sistem operasi buatan Microsoft, selalu update security Patch untuk meningkatkan keamanan sistem operasi tersebut.